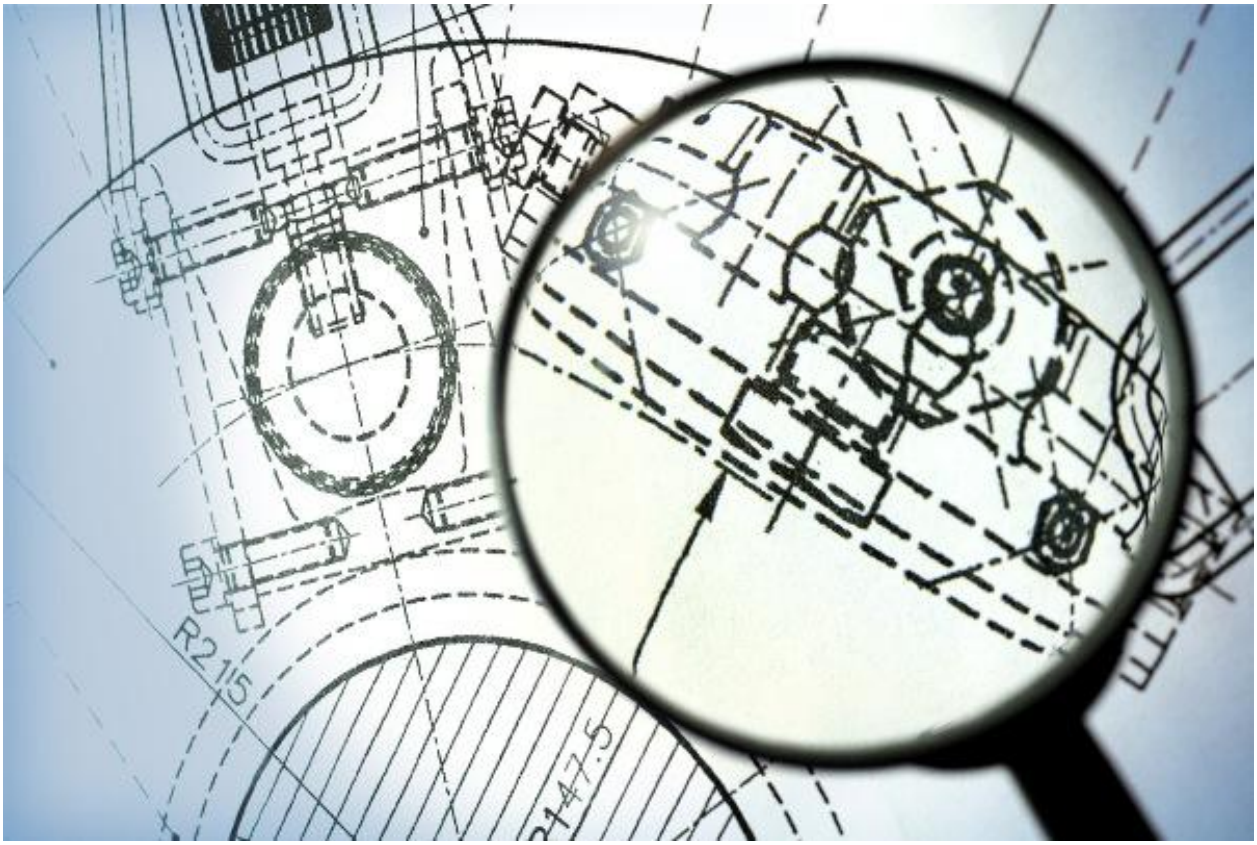
User Manual

Sybit Model Tester



The contents of this document are intellectual property of Sybit GmbH. The document or parts of the document may only be published or passed on to third parties with the permission of Sybit GmbH.

Sybit GmbH
Sankt-Johannis-Str. 1-5
78315 Radolfzell, Germany

Phone: + 49 (77 32) 95 08-0
Fax: + 49 (77 32) 95 08-111

info@sybit.de
www.sybit.com

Dokumentenart	Benutzerhandbuch EN
Dokumenttitel	Sybit Model Tester
Dokumentuntertitel	Benutzerhandbuch
Verfasser	Sybit GmbH Sankt-Johannis-Str. 1-5 78315 Radolfzell
Dateiname	SMT_Benutzerhandbuch_EN_2.07.doc
Model-Tester-Version	2.0
Dokumentenversion	2.07
Status	Freigegeben
Freigabedatum	21.06.2010

Contents

1 Introduction 5

2 Basic Concepts 5

2.1 Example 6
2.2 Test Case and Test Run 6
2.3 Partial Tests vs. Complete Tests 8
2.4 Test Suite 9
2.5 Profile 9
2.6 SAP Engineering Change Management (ECM) 9

3 First Steps 10

3.1 Create a Test Case 10
3.2 Run the Test Case 10
3.3 Evaluate 11
3.4 Further Steps 11

4 Reference 12

4.1 Common Header Area 12
4.2 Common Footer Area 15
4.3 Navigation, Loading, and Personal History 16
4.4 Common Selection Component for Test Cases and Test Suites 17
4.5 Welcome 21
4.6 Import 22
4.7 Test Case 26
4.8 Test Suite 37
4.9 Test Run 39
4.10 Evaluation 40
4.11 Mass Maintenance 46
4.12 Profile 49
4.13 Job 55
4.14 Saving Test Cases from CU50 59

5 Error Handling 61

5.1 Termination (Dump) of the Application 61
5.2 Wrong Functionality 61

Appendix 62

A. Messages 62

A.1. VC-Specific Messages 62
A.2. Errors in the Test Case Structure 63
A.3. Error Messages Caused by Testing Prices 63

B. Further Information 64

1 Introduction

This document serves as a user manual and technical reference for users of the Sybit Model Tester (SMT). It is organized as follows.

Chapter 2 covers the basic concepts of the tool, such as test case and test run. Chapter 3 contains a short tutorial that takes only a few minutes.. It shows – on a beginner level – how to create, execute, and evaluate a test. Chapter 4 contains a detailed reference of all available functions. Finally, Chapter 5 tells you what to do, if you encounter malfunctions in the application.

SMT 2.0

In this manual, features that have been introduced or fundamentally changed in this version are marked with a label on the side margin. An example is displayed here.

2 Basic Concepts

The Sybit Model Tester is a workbench to create, save, organize, and execute test cases in the SAP configuration area. It replaces manual testing – e.g. via transaction CU50 – to a large extent. The basic idea of the Sybit Model Tester is to save test cases and to make them automatically reproducible at any time.

Hence, the central element of Sybit Model Tester is the **test case**.

A test case consists of two parts. The first part is the **input data.** Here, you specify the user input for the test case. The second part is the **expected result**. The expected result contains the user's expectation on what the variant configurator (LO-VC) should do based on the input data.

If you run a test case – which is then called a **test run** – the Model Tester hands over the input configuration to the LO-VC. Subsequently, the LO-VC computes a result which we call the **factual result**. The Model Tester automatically compares the expected with the factual result. If the two results match then the test run is successful (denoted by a green light). If the expected result differs from the factual result then the test run fails (denoted by a red light). The behavior of test runs (i.e., its evaluation) can be influenced by means of so-called **profiles**.

Fig. 2-1 (a) shows the traditional way of manual testing via CU50. The tester manually compares the factual result that is computed by LO-VC with the testers expected result. Fig. 2-1 (b) shows the scheme in a Sybit Model Tester scenario. Here, the expected result is stored inside the test case and – hence – can be compared automatically with the factual result.

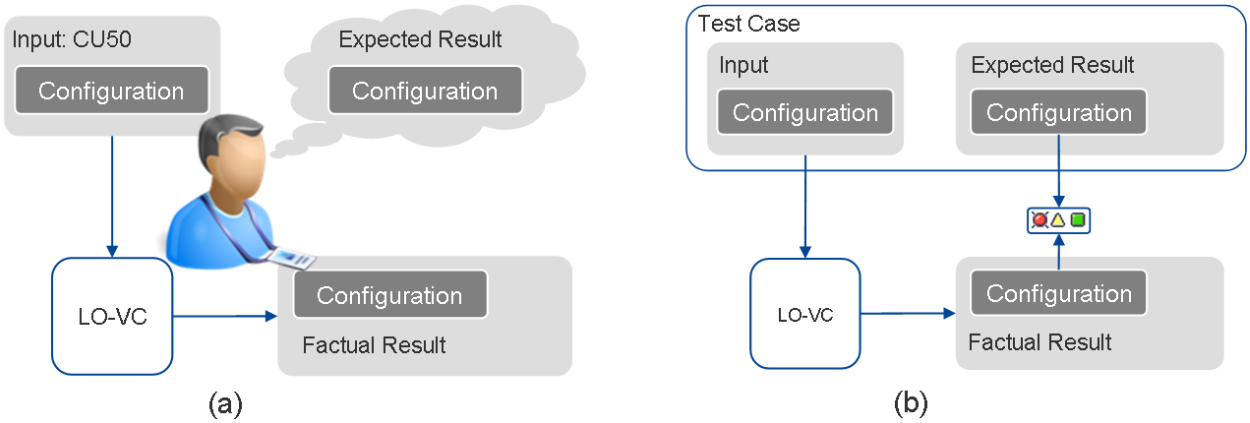


Fig. 2-1: Testing of configuration models. (a) Manual testing. (b) Scenario with the Sybit Model Tester.

2.1 Example

A product has characteristics for size and color. The configuration model specifies that very large instances of the product (`Size = XXL`) should only be available in red (`Color = Red`). Hence, if you input `Size = XXL` the model should automatically infer `Color = Red`. A test case for this behavior would look like this:

Input configuration: `Size = XXL`

Expected result:
```
Size  = XXL
Color = Red
```

The expected result also contains the value `XXL` for characteristic `Size`, since the LO-VC does not delete the value.

When the test case is executed, the LO-VC generates a factual result that is recorded by the Model Tester. The following table shows several cases of potential factual results and how the Model Tester deals with them.

Factual Result	**Comment**
`Size = XXL` `Color = Red`	Factual result matches the expected result: Test run successful, green light
`Size = XXL` `Color =`	Color has not been set: Test run not successful, red light
`Size = XXL` `Color = blue`	Color is wrong: Test run not successful, red light
`Size =` `Color =`	XXL could not be set: Test run not successful, red light

Table 2-1: Alternatives of different factual results for the example.

2.2 Test Case and Test Run

A test case covers different types of tests.

- **Configuration**: The most frequent test type checks if the configuration model sets valid characteristic values. On input of a configuration (i.e. a set of characteristics and values) the

LO-VC computes an inferred configuration that is deduced from the input configuration by means of the configuration model. To this end, the Model Tester's expected result contains an area to specify an expected configuration.

- **Bill of materials (BOM)**: The expected result contains an area to specify the expected bill of materials, to test the BOM explosion.
- **Negative tests**: An important use case is to find out if certain configuration instances are prohibited by the configuration model. In the manual testing scenario, these conflicts are displayed inside a popup dialogue. In the Sybit Model Tester, the expected result contains an area where you can specify error and warning messages. If an expected message is not generated by LO-VC, i.e. if the factual result does not contain the message, the test run fails.

SMT 2.0

In older versions of the Sybit Model Tester, the area for expected messages is called "expected errors". Since it has caused some confusion in combination with errors in the test run, this term has been changed.

SMT 2.0

- **Prices**: If there are errors in pricing, financial damage is often unavoidable. You can test condition records that are controlled by variant conditions in the prices area.

Fig. 2-2 gives an overview including the test case (with input configuration and expected result), the LO-VC, and the factual result.

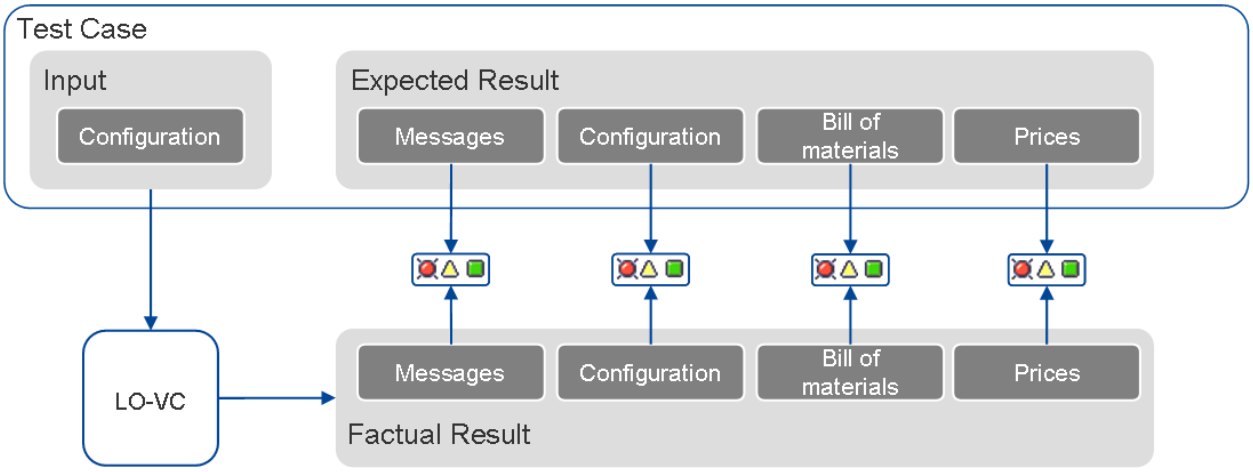


Fig. 2-2: A test case consists of an input configuration and an expected result. When a test is run, the input configuration is passed to the LO-VC that generates a factual result. Finally, the Model Tester compares the factual result with the expected result.

The comparison between expected and factual results is executed for each area separately, i.e. messages, configuration, bill of materials and prices. To give a quick overview of these results, the Sybit Model Tester uses icons that have the colors of traffic lights. The meaning of the colors is as follows:

- **Red light:** The test run produced an error.
- **Yellow light:** The test run produced a warning
- **Green light:** The test run produced neither an error nor a warning
- **Grey light**: Nothing has been tested. This case occurs, if the expected and the factual result are both empty, or if the area is marked as irrelevant by a profile.

2.3 Partial Tests vs. Complete Tests

The Sybit Model Tester focuses on the execution of regression tests of configuration models – i.e. periodically recurring test runs. Hence, it supports the creation and maintenance of configuration models over their entire life cycle. To realize effective quality control, you have to make sure that the following two aspects are met:

1. All values from the expected result are contained in the factual result.
2. All values from the factual result are contained in the expected result.

In the Sybit Model Tester, the standard test case tests both aspects. The run of such test case is successful if and only if the expected result exactly matches the factual result. (Here, we neglect effects that you may cause using profiles – for this, please refer to Section 2.5.) In the Sybit Model Tester, this standard test type is called **complete**. It can be identified by the following icon:

However, during the implementation phase of a configuration model we often have the situation where we want to test *specific aspects* of a configuration model, although not all aspects work correctly. To support this kind of testing, you can define **partial** tests in the Sybit Model Tester. You can identify partial tests by means of the icon. Partial tests only cover the first of the two above aspects, i.e. if all values from the expected result are contained in the factual result. If there are values in the factual result that you did not expect, the test run is successful anyway.

Table 2-2 summarizes the differences between a test run of a partial test and the test run of a complete test.

Area	Partial Test Case	Complete Test Case
Messages	- Only those messages are shown that affect characteristic values contained in the input configuration. - No messages from structure conflicts in multi-level configuration - No messages concerning variant conditions that are not contained in the expected result.	All messages are shown.
Configuration	- Characteristic values that are missing in the expected result **are not** considered as a failure. - Missing (or incorrect) characteristic values in the factual result **are** considered as a failure. - The factual result only contains characteristic values that are contained in the expected result.	- Characteristic values that are missing in the expected result **are** considered as a failure. - Missing (or incorrect) characteristic values in the factual result **are** considered as a failure. - The factual result contains **all** characteristic values.
Bill of Materials (BOM)	- Missing entries in the expected result **are not** considered as a failure. - The factual result only contains entries that are contained in the expected result. For the comparison between expected and factual result, all properties (that are not marked as irrelevant by the profile) of the BOM line are used.	- Missing entries in the expected result **are** considered as a failure. - The factual result contains **all** entries.
Pricing	- Missing condition records in the expected result **are not** considered	- Missing condition records in the expected result **are** considered as

	as a failure. - The factual result only contains condition records that are contained in the expected result.	a failure. - The factual result contains all condition records.

Table 2-2: Differences between partial and complete test cases

2.4 Test Suite

To simplify the management of test cases, they can be organized in test suites. A **test suite** is a container that may contain any number of test cases. Those test cases can be of any type, see Section 2.3. One single test case can be contained by multiple test suites.

2.5 Profile

In the Sybit Model Tester, you may use **profiles** to fine tune the behavior of test runs. Depending on the usage scenario it might be useful to disregard certain information from the configuration area or the BOM area, i.e. if an error occurs in an irrelevant part, the user should not be informed. In the profile you may specify which areas should be relevant for the evaluation. In particular you may even define that you are not interested in the values of certain characteristics. This is helpful, for example, for invisible characteristics that are only used for internal purposes. In the BOM area you may define whether the evaluation should use all available information or if, i.e., inconsistencies in position numbers or units should not be considered relevant to the user. If you don't use a profile for a test run, all available information is considered relevant.

2.6 SAP Engineering Change Management (ECM)

SMT 2.0

Until version 1.5, the Sybit Model tester could be used in two different variants which were targeted towards enterprises with and without the SAP Engineering Change Management (ECM). The variant without ECM was dispensed with in version 2.0. If you want to get rid of the according elements in the user interface, you can do this by means of the Web Dynpro standard personalization functionality. We are happy to support you in this respect, see Appendix B.

3 First Steps

This chapter contains a first tutorial. Here you can create, execute and evaluate your first test. We expect the tutorial to take about 10 minutes.

3.1 Create a Test Case

- **Open**: Open the Sybit Model Tester
- **Insert KMat**: In the header area, there is a field labeled **KMat** that contains the ID of the configurable material under test. You may enter a material directly or use the value help. To begin with, you may want to choose a simple material with only few required characteristics.
- **Test case**: Navigate to the tab **Test Case**.
- **Create**: Click on **New**.
- **Name / Description**: Insert a name and a description for the respective input fields, i.e. “first test” and “Tutorial”.
- **Input configuration**: The main work area consists of a navigation table and two areas for input configuration and expected result. The table for input configuration should contain all characteristics of the chosen KMat. Please enter a valid configuration into the table (similar to CU50).
- **Expected Configuration**: The area on the right contains the expected result. Switch to the tab Configuration.
- **Generate Proposal**: In the configuration table of the expected result you could enter the expected configuration manually. A more convenient way to create the expected result is to let the LO-VC generate the result and only interfere if this is not what you actually expect. To achieve this, take a look at the toolbar above. Besides the label **Expected Result** there is a checkbox for each area of the expected result. Check the check boxes for **M** and **C** and click on **Generate Proposal**. Now the LO-VC fills the message and the configuration area of the expected result with the output of the current configuration model. If your input configuration is invalid you will see a warning inside the message area at the very bottom of the screen. If you need details on the errors just switch to the messages tab of the expected result. Then, correct your input configuration and press generate proposal again. Repeat these steps until you have no messages left in the message tab.
- **Save**: Click on the save button.

3.2 Run the Test Case

- **Test Run Tab**: To switch to the Test Run tab you may use the **Quick Link** at the lower right of your screen. You are taken directly to the test run tab where your test case is already loaded and selected inside the selection table. As an alternative you can also click on the test run tab and search for your test case in the data base.
- **Execute**: Click run. On successful execution you will see a message inside the message area.

3.3 Evaluate

- **Evaluation Tab**: Switch to the evaluation tab once again by using the **Quick Link** at the lower right of your screen. The executed test run is already loaded and selected in the evaluation table. As an alternative, you may also click on the evaluation tab directly and load your test run.
- **Analysis**: A click on the **Analyze** button opens the analysis popup. You can see the expected result on the left and the factual result on the right hand side for all areas. If everything went o.k. you should only see green or grey lights.

3.4 Further Steps

- **Test case**: A click on the name of the test case brings you back to the test case tab. There you may change the input configuration or the expected result.
- **Test run**: Switch to the test run tab and execute your test case.
- **Evaluation**: Check out how your changes affected the result of the test run.

4 Reference

The user interface of the Sybit Model Tester consists of a number of tabs that share a common header and footer area. In this section we first present the header and footer area and a common user interface component to select test cases and test suites. Subsequently, we go into detail for each single tab.

4.1 Common Header Area

The header contains a number of fields that depend on your enterprise specific application configuration (Fig. 4-1). If your scenario is targeted to test multiple LO-VCs in different systems via RFC, there is a dropdown box labeled **RFC Dest.** that contains all possible connections. If your scenario is targeted to one instance of the LO-VC the dropdown box is invisible.

The **KMat** field contains the ID of the configurable material (KMat) under test. The **Date** field contains the logical model date. Example: if you enter 08.11.2010 in the date field then your test runs will be executed with the configuration model that is active at that date.

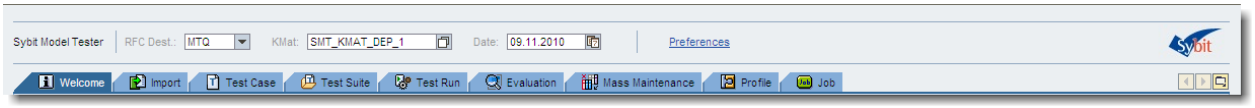


Fig. 4-1: Common header.

If you click on **Preferences** you will get a popup that contains system settings and personal preferences (Fig. 4-2). The navigation tree on the left contains the following chapters: **System Information**, **Presentation**, **Application Parameters**, **Testcase Parameters,** and **Statistics and License**.

When you open the popup, the **System Information** chapter is opened (Fig. 4-2). It contains four sections. The section **General** gives you information about your user and the system you are working on. In the section **Application Log** you can display the application log and switch it on and off. You should be aware that switching on the application log can have negative impacts on your application performance. The section on **System Configuration** contains information on your enterprise specific configuration of the Sybit Model Tester application. Here you can view the RFC destinations that are available in your scenario as well as a list of users who have administration privileges. Your personal settings which the system will remember until your next session are displayed in the **Personalization** section.

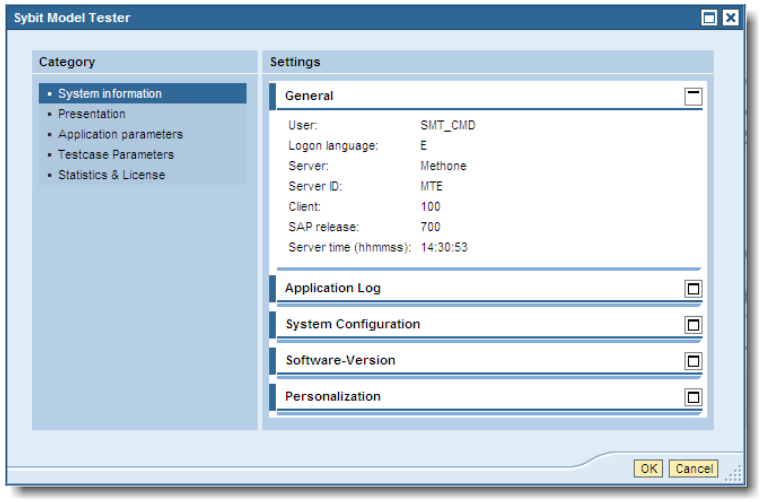


Fig. 4-2: Preferences popup: Chapter System information

In the chapter **Presentation**, you can switch between language dependent and language independent display of master data (Fig. 4-3). Here we distinguish between the presentation of characteristics and their values (upper part of the view), and the presentation of KMats in the navigation tables in the test case or the profile tab (lower part of the view). If you chose “both options”, both texts are displayed side by side. You may only chose this setting for characteristics and their values.

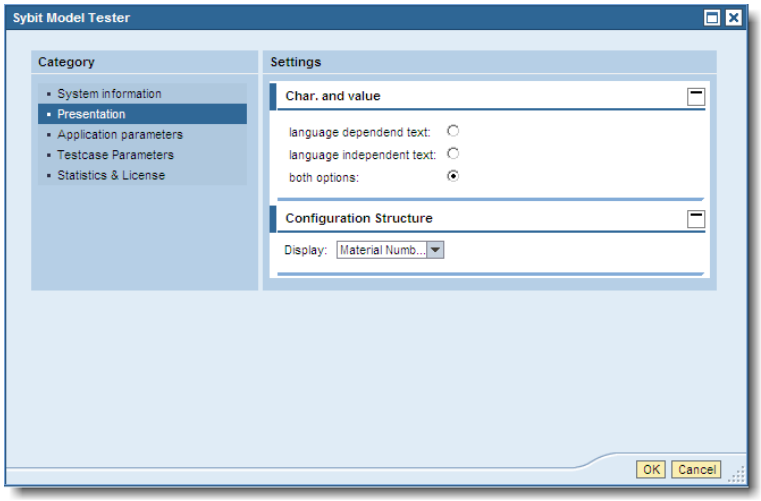


Fig. 4-3: Preferences popup: Chapter Presentation.

You can set the application size in the **Application Parameters** chapter (Fig. 4-4). Here you can optimize the layout of the application to your screen resolution. If you click on the Reset Default Values button the application size is reset to the default values of 1200x800 pixels. In the lower section you can customize the application to always set the current date into the Date field in the header, or to remember the date and set it in your next session.

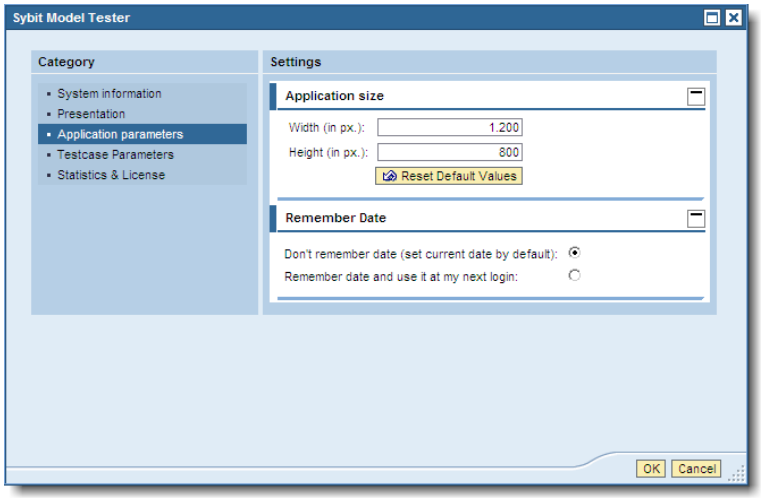


Fig. 4-4: Preferences popup: Chapter Application parameters.

You can set the following preferences in the chapter on **Testcase Parameters** (Fig. 4-5):

General Settings:

If you remove the check from "refresh automatically while 'generate proposal'", then the generate proposal button in the test case tab changes its behavior. As an illustration, consider the following two workflows that have the same result:

Workflow 1	**Workflow 2**
• Preferences: Check is set • Test case tab: Click on generate proposal	• Preferences: Check is **not** set • Test case tab: Click on generate proposal • Test case tab: Click on refresh

Default Settings for the bill of materials:

- **Plant:** If you create a new test case in the test case tab, then this plant is inserted into the new test case.
- **Configuration:** If you create a new test case in the test case tab, then this BOM configuration is inserted into the new test case.
- **Application**: If you create a new test case in the test case tab, then this BOM application is inserted into the new test case. Choose Automatic if you want the BOM application to be determined by the configuration profile. You have to chose Manual and set the according application only if there is no BOM application in the configuration profile.
  This parameter is not only used when creating test cases in the test case tab, but also when you import test cases from sales documents.

SMT 2.0

Default Settings for prices

- **Calc. Schema:** If you create a new test case in the test case tab, then this calculation schema is inserted into the new test case. If you don't want to test pricing, just leave the field empty. Then all test cases will be created with empty calculation schema.

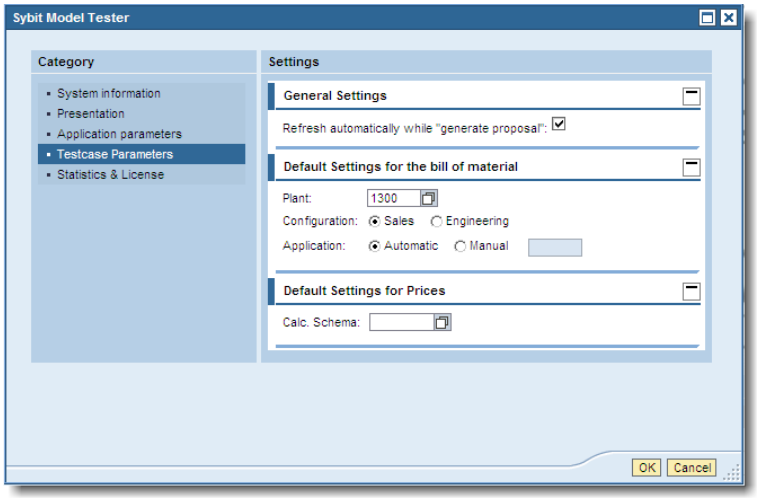


Fig. 4-5: Preferences popup: Chapter Testcase Parameters.

The chapter on **Statistics & License** (Fig. 4-6) provides you with some statistics on the usage of your installation of the Sybit Model Tester. This also includes information about the number of licensed characteristics and how many you already test. Caution! The computation of these statistics causes complex database queries that may take a while.

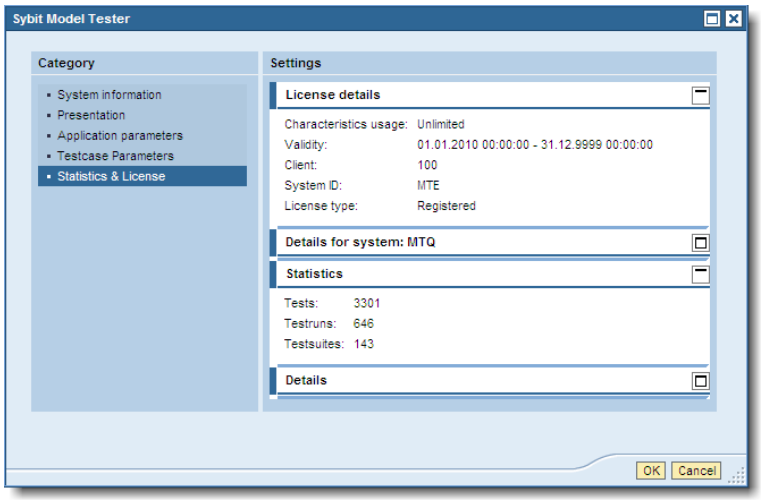


Fig. 4-6: Preferences popup: Chapter Statistics & License.

4.2 Common Footer Area

The footer area comprises two parts (Fig. 4-7). On the left side you find the message area of the application. Here, different types of messages may occur. Informative messages give a positive feedback on system actions. They come along with a green light. Warning messages are yellow. Red messages occur in cases where the system could not execute out your requested action.

On the right side you find the navigation area containing so-called “quick inks”. Depending on your actions the system suggests where you could go next. Example: if you are viewing a test case inside the test case tab, there is always a quick link to the test run tab.

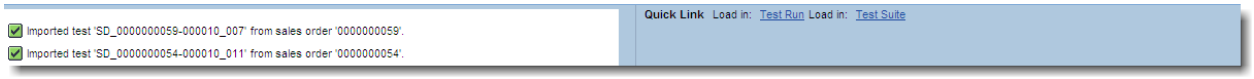
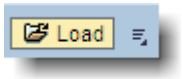

Imported test 'SD_0000000059-000010_007' from sales order '0000000059'.
Imported test 'SD_0000000054-000010_011' from sales order '0000000054'.
Quick Link Load in: Test Run Load in: Test Suite

Fig. 4-7: Common footer area.

4.3 Navigation, Loading, and Personal History

In the Sybit Model Tester, navigation and loading of objects can be achieved by different GUI elements.

- **Welcome tab:** In the welcome tab (see Section 4.5) there is a table that displays your recently executed actions that you may wish to resume. These actions depend on the settings in the header area. You will find different actions in this table for different KMats, see Table 4-1.
- **Click on tabs:** Clicking on tabs is the classical way of navigation. You can do this at any time and in any state of the application. If there are unsaved changes in your current working tab, you will get a popup that asks if you want to save or discard those changes before navigation.
- **Load button:** The following tabs contain buttons to load an object: test case, test suite, evaluation, mass maintenance, and profile. If you click the button, a popup will appear, where you can search for and subsequently load found objects. Search results depend on the information on KMat, date, and RFC-Dest that are specified in the header, see Table 4-1.
- **Personal history:** In the following tabs you may load objects from your personal history: test case, test suite, evaluation, and profile. To do this, click on the small symbol for drop down lists beside the load button.

  Objects shown here depend on the information on KMat, date, and RFC-Dest that are specified in the header, see Table 4-1.
- **Navigation area in the footer**: In many tabs, you will get navigation helps by means of quick links in the footer area, see Section 4.2.

Tab	RFC-Dest.	KMat	Date	Explanation
Welcome		X		Has effect on the personal history table.
Test Case		X	X	Only the following test cases are found: - they test the given header KMat - their validity time span contains the header date.
Test Suite		X		Test suites are specific to KMats
Test Run	(X)	X	X	The test run is executed in the system that is specified in the RFC-Dest. If you search for test cases, KMat and Date are respected. If you search for test suites, only the KMat is respected.

Evaluation	X	X		If you search for test runs, you will get only runs in the header destination system by default. However, you may change this default inside the load popup.
Mass Main-tenance		X	X	Same as in Test Case
Profile				Loading profiles disregards data from the header.
Job				Loading jobs disregards data from the header.

Table 4-1: Influences of the information in the header on loading functionality in each tab. The RFC-Dest. is visible in the header only if you have configured multiple target systems in your scenario.

4.4 Common Selection Component for Test Cases and Test Suites

There is a common visual component to select test cases and test suites at various places throughout the application, see Fig. 4-8.

The component comprises two tabs labeled with **Tests** and **Suites.** You can search and filter test cases directly in the tests tab. You may use the suites tab to take advantage of test suites as a means to organize test cases. Example: you can search for test suites and, subsequently, you may choose single test cases from within the suites. Since the functionality in the suites tab is similar to the tests tab, we only discuss the tests tab.

The **Standard** search comprises fields for **Name** and **Description** of test cases. You may use the wild-card character ***** to perform a fuzzy search. Please check the respective check box to make a case sensitive search. To reduce the size of your result you can restrict the **maximum number of hits**. This also reduces the running time of your search query.

If you open the **Extended** section (Fig. 4-9) you can set additional filters for:

- **Created by**: The login name of the user who created the object.
- **Created at (from / to):** The time span in which the object has been created.
- **Last changed by**: The login name of the user who made the last change to the object.
- **Last changed at (from / to)**: The time span of the last change to the object.
- **Type**: Here you may filter partial or complete tests, see Section 2.3.

If you open the **Last Run** section (Fig. 4-10), you can set additional filters for:

- **Last Run by**: Login name of the user who executed the last test run.
- **Last Run at (from / to)**: Time span in which the last test run has been executed.
- **RFC-Dest:**: The target system of the last run
- **Model Date:** The logical date of the configuration model that has been used for the last run.
- **Last Result:** The result that has been achieved in the last run. The result is split into messages, configuration, BOM, and prices.

If you open the **Characteristics** section (Fig. 4-11), you can set additional filters for characteristic values. Example: all resulting test cases contain the given value for the given characteristic. In the current version, not all types of characteristics are supported here.

If you **Search** for test cases, you will only find those test cases that contain the global date within their validity period.

The button **Number of Entries** opens a popup which displays the number of objects that comply with your filter criteria. If this number is very large, you should increase the restrictions of your filter. The **Reset** button deletes all filter entries.

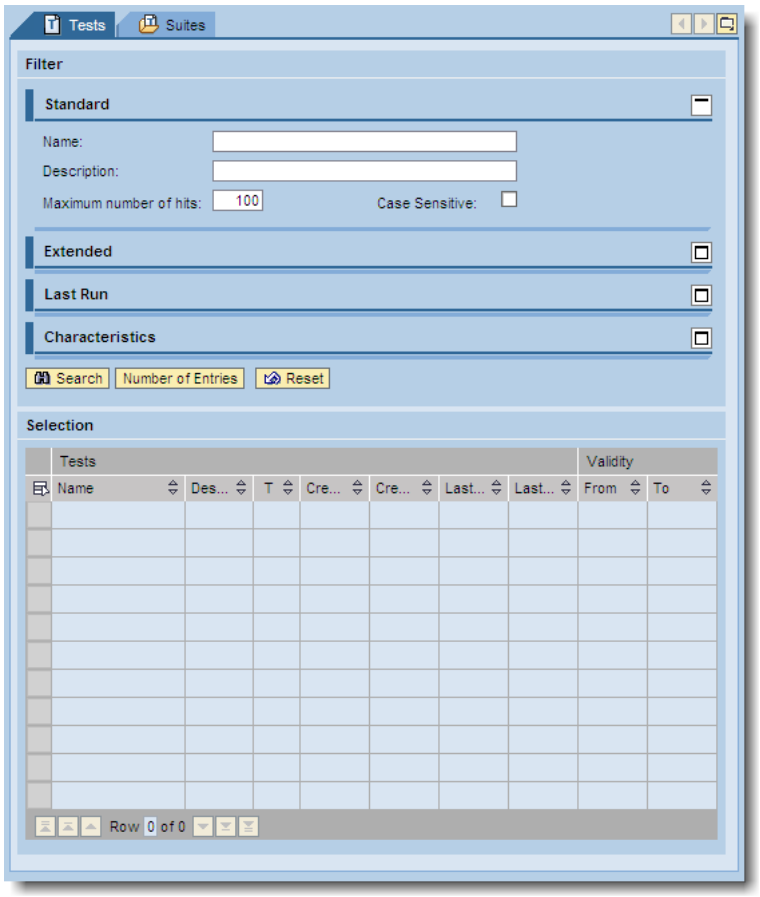


Fig. 4-8: Selection of test cases and test suites.

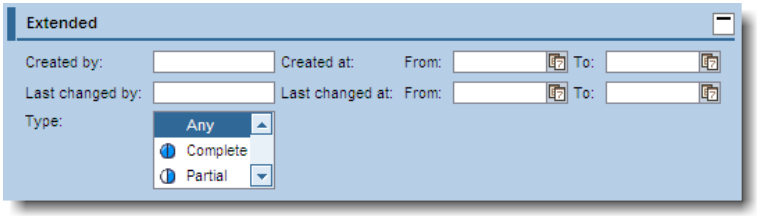


Fig. 4-9: Extended search options.

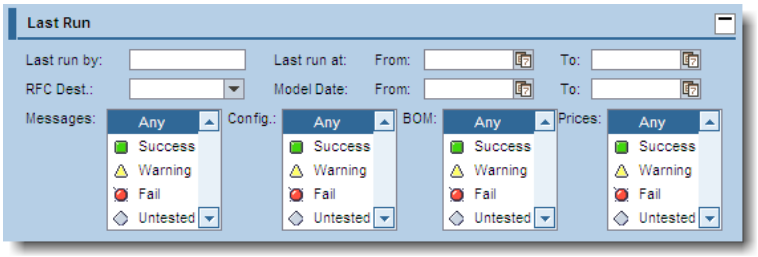


Fig. 4-10: Search options relating to the last run.

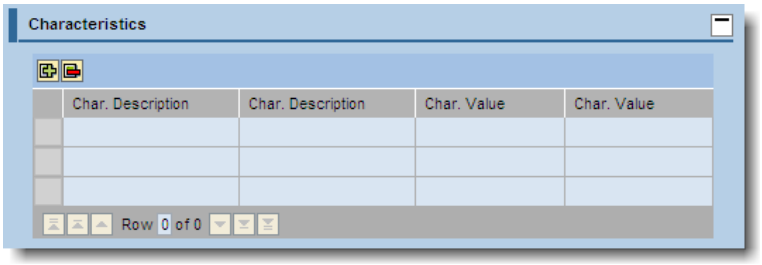


Fig. 4-11: Search options related to characteristic values.

You can display the following table columns in the selection area, see Fig. 4-12.

- Column group **Tests**
  - **Name**: Name of the test case
  - **Description**: Description of the test case
  - **Type**: Icons for complete and partial test cases
  - **Created by**: Login name of the user who created the test case
  - **Created at:** Time stamp of the creation time
  - **Last change by**: Login name of the user who made the last change
  - **Last change at:** Time stamp of the last change
- Column group **Last run at**
  - **Last run by**: Login name of the user who executed the last test run
  - **Date and Time:** Time stamp of the last run
  - **Model Date**: Logical model date with which the run has been executed
  - **RFC Dest.**: System in which the run has been executed
- Column group **Last Result**
  - **M**: Icon for the result of the last run: Messages area
  - **C**: Icon for the result of the last run: Configuration area
  - **B**: Icon for the result of the last run: Bill of materials area
  - **P**: Icon for the result of the last run: Prices area
- Column group **Validity**
  - **From**: Start of the validity time span of the test case
  - **To**: End of the validity time span of the test case

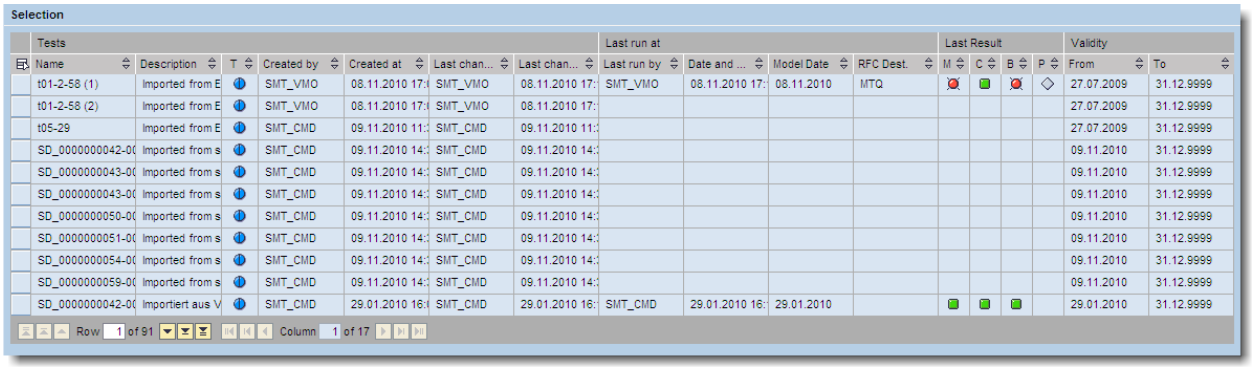


Fig. 4-12: Selection area for test cases.

Screen space is restricted and most often you don't need to see all columns of the table. To focus on information that is relevant to you, you can use the following two mechanisms:

- **Horizontal paging**: If the there is not enough space in the application to display all table columns, you may scroll horizontally over the columns of the table. To this end, there are horizontal paginators at the bottom of the table. With those small buttons, you may scroll to the left or to the right.
- **Hiding columns:** You may hide columns or column groups to suit your personal requirements. To achieve this, position your mouse pointer on the header of the column or column group that you want to hide. Then right-click to open the context menu, see Fig. 4-13. Chose **User Settings → Hide Table column**. Via **User Settings → invisible elements** you can re-display hidden elements.
  The application will remember hidden table columns in your personal profile. Next time you start the application hidden columns will still be invisible. You may personalize the table differently for each occurrence. Example: if you load a test run for evaluation, you may want to see the result of the last run. Whereas if you load a test case in the test case tab, you may want to see the name of the user who made the last change.

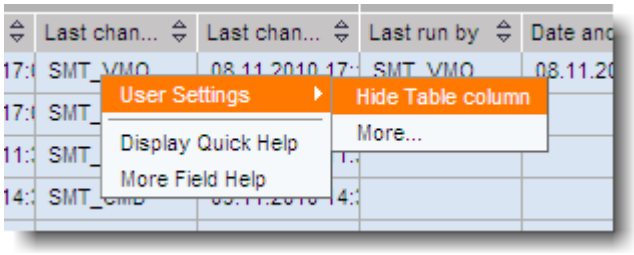


Fig. 4-13: Hiding table columns using the context menu.

It is important to understand, that test cases may be run independently from test suites. Hence, it is possible that the last result of a test suite is green where at the same time the suite contains a red test case. This situation occurs if the test case was green when the suite has been run. In a subsequent run, however, the test case failed and hence has a red icon.

4.5 Welcome

The welcome tab gives general hints for the application (Fig. 4-14). It contains hyperlinks to the following information sources:

- Website with current version of this user manual.
- Sybit Model Tester support email address.
- Sybit Model Tester homepage
- Software version and Sybit homepage

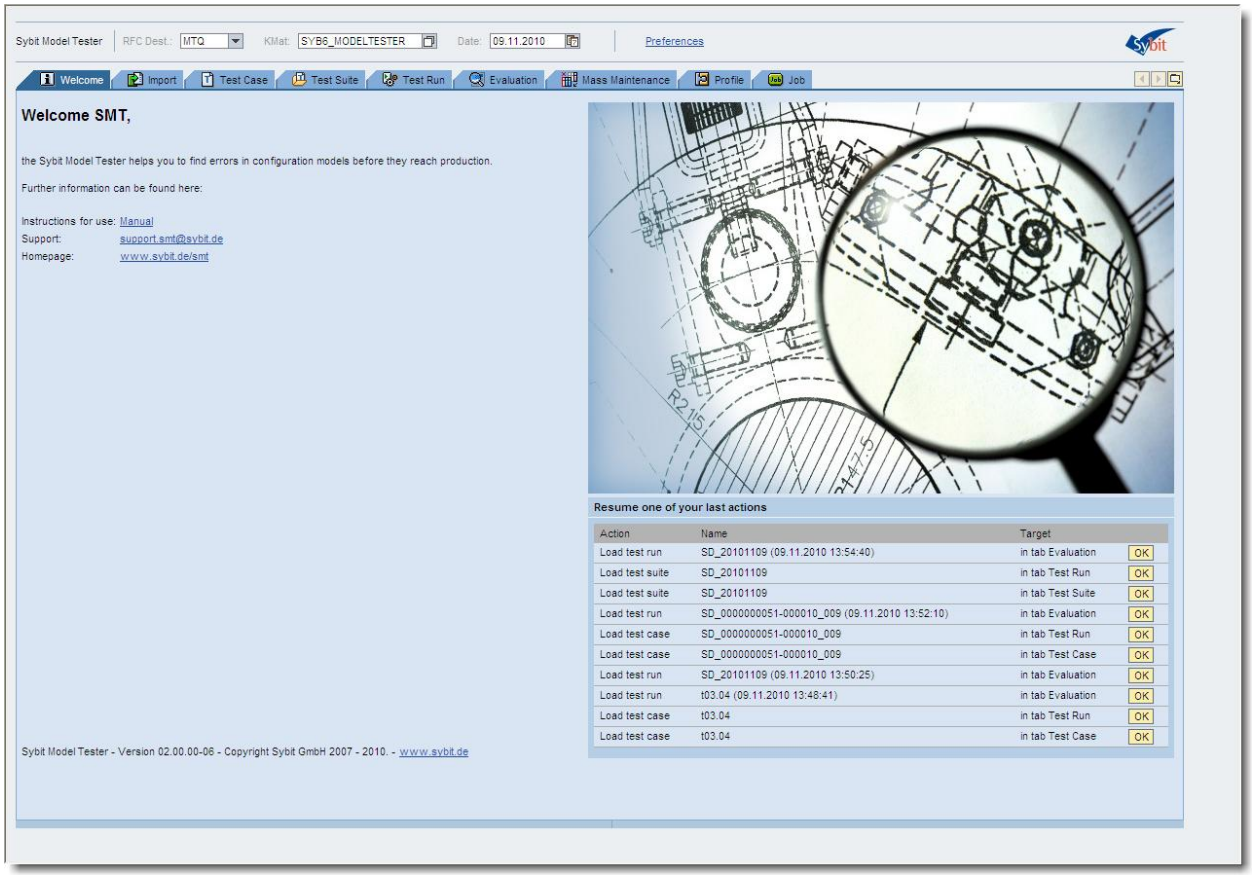


Fig. 4-14: Welcome tab

Additionally, the welcome tab enables you to resume your previous work. The table below the image contains your personal history where your most recent actions are listed at the top.

Each table line offers you the possibility to jump into a specific tab and load the object you where working on, at the same time. Here, the displayed actions depend on the the KMat that is specified in the header area, see Section 4.3. Example: you will only see test cases in the table that test the KMat specified in the application header.

4.6 Import

The import tab contains two tools for automatic import of test cases. Each tool resides in a sub-tab of its own. In the SD tab you can import sales documents, create test cases from them and save them in the Model Tester. In the XML tab you can import test cases from external sources via predefined XML formats.

4.6.1 Sales Documents (SD)

In this tab you can import test cases from sales documents, see Fig. 4-15. To realize this, the Model Tester accesses the configuration data that is stored inside the document. User input values from the sales document are set as input data of the test case. User input values and system values are set as the configuration part of the expected result. The bill of materials of the test case is generated from the bill of materials from the sales document.

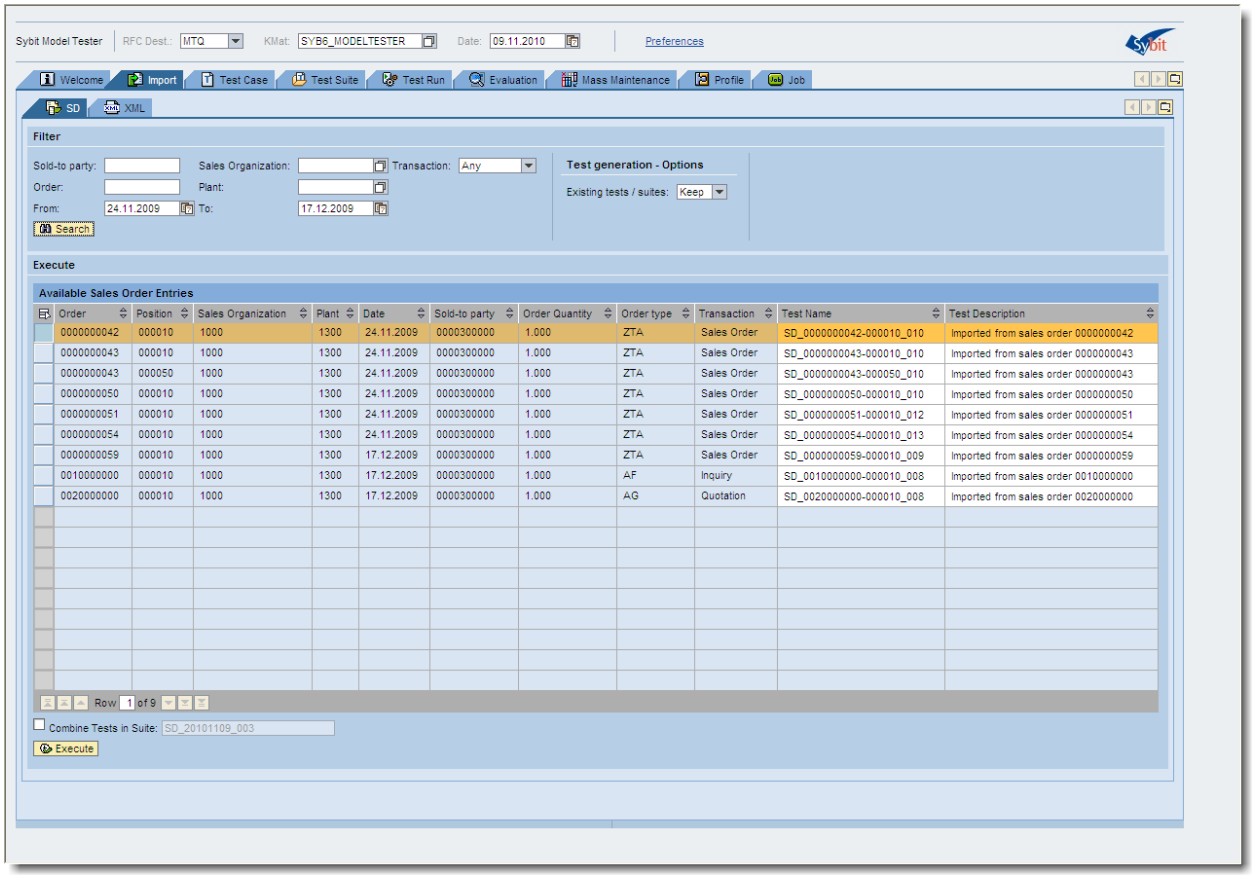

Sybit Model Tester
RFC Dest.: MTQ
KMat: SYB6_MODELTESTER
Date: 09.11.2010
Preferences
Welcome
Import
Test Case
Test Suite
Test Run
Evaluation
Mass Maintenance
Profile
Job
SD
XML
Filter
Sold-to party:
Sales Organization:
Transaction: Any
Order:
Plant:
From: 24.11.2009
To: 17.12.2009
Search
Test generation - Options
Existing tests / suites: Keep
Execute
Available Sales Order Entries
Order
Position
Sales Organization
Plant
Date
Sold-to party
Order Quantity
Order type
Transaction
Test Name
Test Description
0000000042 000010 1000 1300 24.11.2009 0000300000 1.000 ZTA Sales Order SD_0000000042-000010_010 Imported from sales order 0000000042
0000000043 000010 1000 1300 24.11.2009 0000300000 1.000 ZTA Sales Order SD_0000000043-000010_010 Imported from sales order 0000000043
0000000043 000050 1000 1300 24.11.2009 0000300000 1.000 ZTA Sales Order SD_0000000043-000050_010 Imported from sales order 0000000043
0000000050 000010 1000 1300 24.11.2009 0000300000 1.000 ZTA Sales Order SD_0000000050-000010_010 Imported from sales order 0000000050
0000000051 000010 1000 1300 24.11.2009 0000300000 1.000 ZTA Sales Order SD_0000000051-000010_012 Imported from sales order 0000000051
0000000054 000010 1000 1300 24.11.2009 0000300000 1.000 ZTA Sales Order SD_0000000054-000010_013 Imported from sales order 0000000054
0000000059 000010 1000 1300 17.12.2009 0000300000 1.000 ZTA Sales Order SD_0000000059-000010_009 Imported from sales order 0000000059
0010000000 000010 1000 1300 17.12.2009 0000300000 1.000 AF Inquiry SD_0010000000-000010_008 Imported from sales order 0010000000
0020000000 000010 1000 1300 17.12.2009 0000300000 1.000 AG Quotation SD_0020000000-000010_008 Imported from sales order 0020000000
Row 1 of 9
Combine Tests in Suite: SD_20101109_003
Execute

Fig. 4-15: Tab for import of sales documents

The following controls are available:

- **Filter area**: Here you can define filter criteria for the sales documents displayed below. There are filters for sold-to party, sales organization, transaction, order, plant, and date.
- **Options**: Here you can specify how the import deals with test cases and test suites that are already present in the system. You may chose among:

- **Keep**: Doesn't touch existing objects. If there is an object in the system that has the same name, the import will be canceled with an error message.
- **Adapt**: Existing test cases / suites are adapted, i.e. overwritten, with the imported information.
- **Delete**: Existing test cases / suites are deleted first. Subsequently, the import creates new objects in the system. If there is a test suite in the system that contains more test cases than you import here, those test cases will no longer be part of the test suite, once you have choosen the delete option. If you chose adapt, however, they are still part of the suite.

The generation of test case / suite names behaves differently, depending on which option you chose here. If you chose **Keep** the system generates unique names, and possibly names of existing objects.

- **Import area**: The table shows all sales documents that match the above filters. If a sales document contains multiple instances of the KMat, each instance is displayed in a line of its own. You can choose the test cases to import by multi-selection.
  - **Test Name / Test Description**: For each test case the application suggests a name and a description that can be overwritten manually. The system generated name contains the number of the sales document and the position where the KMat occurs within the order.
  - **Combine Tests in Suite**: If you check this check-box, all imported tests are combined into a test suite with the specified name. You can overwrite the system-generated name manually.
  - **Execute**: The execute button generates a test case for each selected table

4.6.2 XML

In this tab you can import XML-formatted test cases into the Model Tester, see Fig. 4-16.. You can choose between two different XML formats:

- The SMT XML format directly reflects the Model Tester internal data structure for test cases.
- The Excel XML format is targeted towards an easy to use XML export from MS Excel. Accordingly, this format should be used if you want to create test cases in Microsoft Excel, instead of directly inside the Model Tester.

The specification of both formats is available as .XSD file. Please contact the Sybit Model Tester support, see Appendix B.

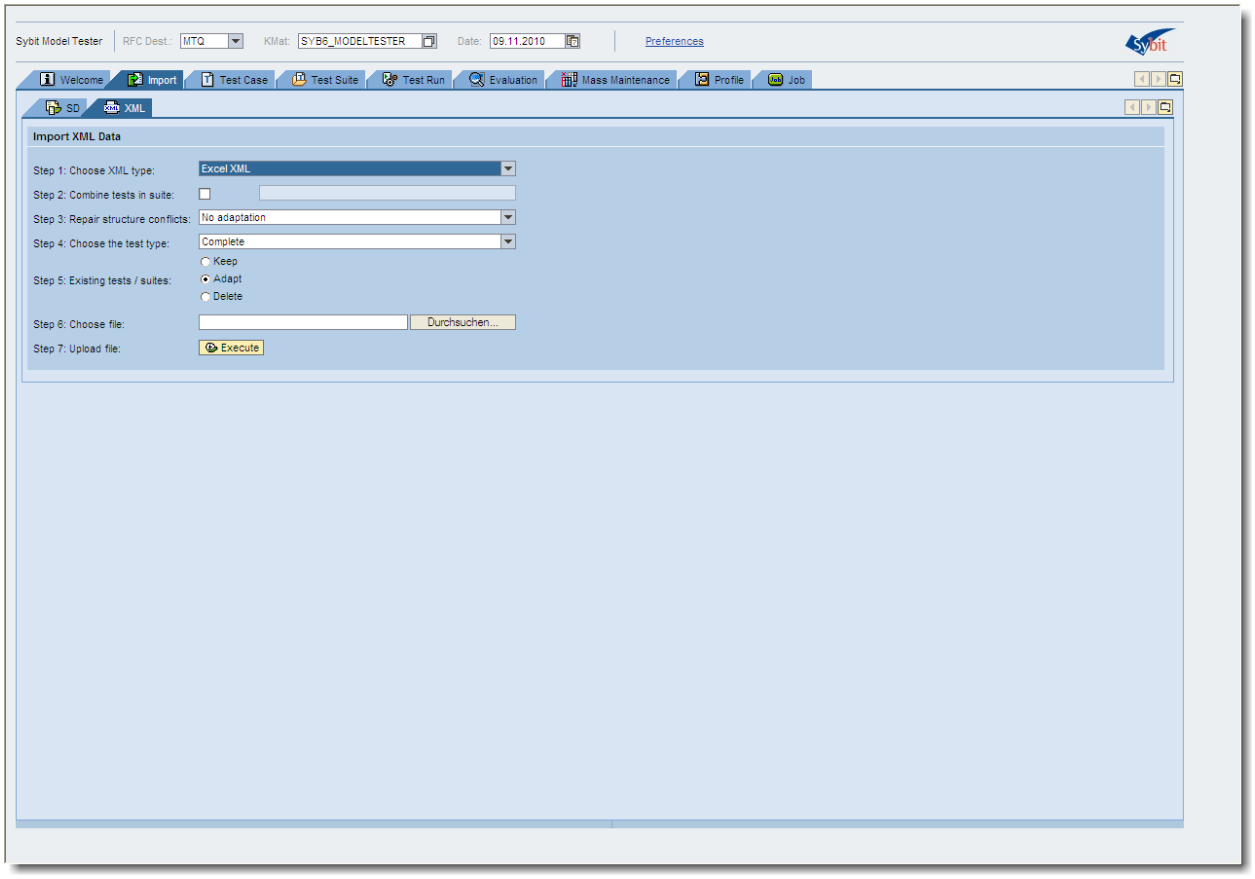


Fig. 4-16: Tab for Import of XML files.

The following controls are available:

- **Chose XML type:** Here, you specify if the import file complies to the SMT-XML or to the Excel-XML format.
- **Combine tests is suite:** This option is only available for Excel-XML. Here you may optionally define the name of the test suite in which all imported test cases should be contained.
- **Repair structure conflicts:** You may chose among the following options:
  - **No adaption:** Test cases are imported exactly as specified in the XML file.
  - **Optimistic:** If you are importing multi-level test cases, the hierarchy structure of the imported test case might not match the current VC model. If you chose optimistic the following actions are executed:
    - **Reassignment:** The system finds an optimal assignment of sub positions in the test case to the current VC model.
    - **Deletion of obsolete sub positions:** Sub positions of a test case that belong to a KMat that is not contained in the present VC model are deleted.
    - **Creation of missing sub positions:** A new sub position is created for all sub KMats in the current VC model that can not be mapped to an existing sub position in the imported test case.
  - **Pessimistic:** Compared to the optimistic option, only Step 1 – Reassignement – is executed. There is no deletion and no creation of sub positions.

- **Choose test type:** Here you can specify whether the imported test cases should be partial or complete, see Section 2.3.
- **Existing tests / suites:** This option tells the system how to deal with naming conflicts, i.e. if objects with identical names already exist in the system. Available options are the same as in the import of sales documents, see Section 4.6.1.
- **Choose file**: Please pick the XML file to import. The file can be uploaded from your local computer or from a shared network drive.
- **Upload file:** Executes the import.

4.7 Test Case

The Test Case tab is used to create and maintain test cases, see Fig. 4-17. The tool is divided in four areas:

- Toolbar with buttons and fields for general test case information (at the top).
- Navigation table for multilevel test cases (left).
- Area for input configuration (middle).
- Area for the expected result (right).

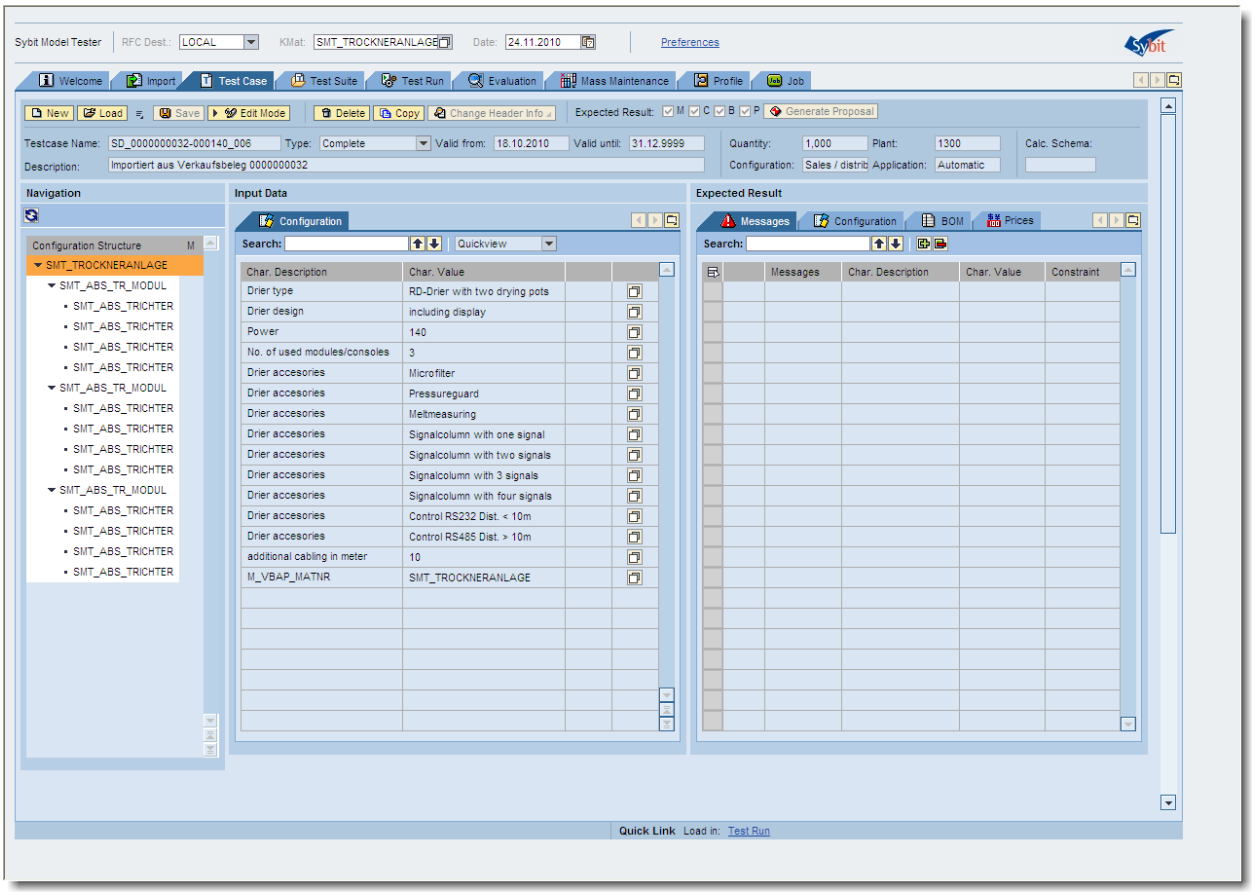


Fig. 4-17: Test case tab: Toolbar, navigation tree, input configuration and expected result.

4.7.1 Toolbar

The toolbar contains the following controls:

- **New**: Creates a new test case.
- **Load**: Loads an existing test case. The load dialog contains the above described component for test case selection, see Section 0.
  Besides the load button, **the history drop down menu** enables you to quickly load test cases from your personal history, see Section 4.3. The objects displayed here depend on the KMat that is specified in the header.

- **Save**: Saves the test case. The name of the test case is not permitted to be empty on save. If you change the test case name before saving the test case, you have then changed the name. This action does not copy the test case.
- **Edit Mode**: Switches from read-only to edit-mode and back.
  If you switch off the edit mode and have unsaved changes a popup will appear that asks if you want to save the changes, discard the changes, or cancel your action.
  If you switch on edit mode for a multi-level test case but your test case structure does not reflect the configuration structure of the Kmat, a popup will appear where you can choose between the following options, see Fig. 4-18:
  - **Adapt**: The structure of the test case is adapted to the current structure of the KMat. Obsolete sub positions are deleted, new sub positions are inserted.
  - **Copy**: The test case is copied and the copy is loaded.
  - **Repair**: This option adapts the test case structure to the current structure of the KMat. You should use this option if position numbers of sub positions have been changed in the bill of materials which requires a re-assignement of sub positions. The following actions are executed:
    - **Re-assignment**: The system finds the optimal assignment of sub positions of the test case to sub positions in the current KMat.
    - **Deletion of obsolete sub positions**: Sub positions of the test case that can not be matched are deleted.
    - **Insertion of missing sub positions**: Sub positions that are missing are created automatically.
- **Delete** (available only in read-only mode): Deletes the loaded test case.
- **Copy** (available only in read-only mode): Creates a copy of the loaded test case and loads the copy.
- **Change header date:** This control contains the following buttons to change header data of the loaded test case.
- **Change BOM:** Changes plant, application, and configuration for the bill of materials.
- **Change Calc. Schema:** Changes the calculation schema of the test case.
- **Change Validity:** changes the validity time span of the test case.
- **Generate Proposal:** This button helps you to fill the expected result directly from the VC model: It passes the input configuration to the VC and loads the generated result into the checked areas of the expected result. The check boxes labeled with M, C, B, and P represent the areas for messages, configuration, BOM, and pricing, respectively.
  A click on the generate proposal button automatically refreshes the navigation tree on the left. If you don't like this behavior, you can easily switch it off as described in Section 4.1 in the chapter of test case parameters. For performance reasons, sub positions that have just been created do not contain an expected result. To get an expected result for them, just press generate proposal again.
  To give you an immediate feedback of error and warning messages in your test case, the Model Tester displays icons inside the navigation tree on the left. The icons have the following meaning:
  - **Red light:** The test case contains an error message in the expected result of this hierarchy level.

- **Yellow light:** The test case contains a warning message in the expected result of this hierarchy level.
- **No icon:** The expected result contains no messages in this hierarchy level.
- **Grey icon:** This hierarchy level has just been created by clicking on generate proposal. Hence, the expected result on this level is still empty. To fill it, click generate proposal again.

- **Name**: Name of the test case. Empty names are not permitted. Test case names must be unique per KMat. Tests with the same name may exist for different KMats.
- **Description**: Description of the test case.
- **Type**: Type of test, see Section 2.3. You may choose complete or partial. If in doubt always use complete which is also the default.
- **BOM Plant**: Plant for BOM explosion.
- **BOM Configuration**: Configuration for BOM explosion (sales / engineering).
- **BOM Application**: Application for the BOM, e.g. SD01 for Sales, PP01 for production, etc.
- **Valid from / Valid until:** Defines the validity period for the test case. The fields can not be edited directly but only by means of the change validity button.
- **Quantity**: Here you may adjust the quantity of the configured instance in your test case. Most often you will have a quantity of 1.0. This option is mainly relevant for test cases that have been generated from sales documents.
- **Calc. Schema**: If you want to test prices with your test case, you need to define the appropriate calculation scheme here. If your test case is not about prices, leave the field empty. In the test run, this will result in a grey icon.

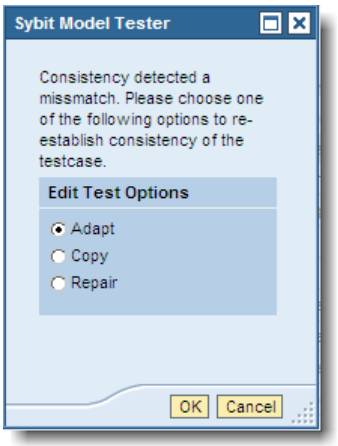


Fig. 4-18: Popup when changing into edit mode with a multi-level test case that has an invalid structure.

4.7.2 Navigation Area

The navigation area is used to change configuration structure levels in test cases for multi-level KMats. In case of multi-level configuration the input configuration and the expected result is displayed and edited for each level separately. To navigate to a different level, just click on the name of the according KMat in the table.

- **Refresh: This button triggers the synchronization of the test case structure with the current** configuration instance inside the LO-VC. If you enter a characteristic value in the input table that provokes, for example, a new sub-KMat then you must press the refresh button to display the updated test case structure. A click on refresh triggers a BOM explosion in the back-end that is based on the current input configuration. The result of the BOM explosion determines the new test case structure.

The icons in the navigation tree have the following meaning:

- **Red light**: The test case contains an error message in the expected result of this hierarchy level.
- **Yellow light**: The test case contains a warning message in the expected result of this hierarchy level.
- **No icon**: The expected result contains no messages in this hierarchy level.
- **Grey icon**: This hierarchy level has just been created by a click on generate proposal. The expected result on this level is empty. To fill it, click generate proposal again.

4.7.3 Input Data

In the input data section you specify the part of the test case that you would type into CU50 if you were in a manual testing scenario (Fig. 4-17).

- **Search**: You can use this input field to search for characteristic names. Arrow up / arrow down searches up / down, respectively. The search engine always scans both, the language dependent and the language independent characteristic names.
- **Quick View / Full View**: Here you may choose between two different view modes in the configuration table. If you choose the quick view (which is the default after loading a test case) you only see characteristics that have a value. If you switch to full view (which you can do in edit-on mode only) you see all characteristics, i.e. also those that have no values. In the full view, the application makes sure that characteristics from input and expected result are always aligned in the table.
  Switching between quick view and full view always affects input and expected configuration.
- **Char. Value column**: Direct input of characteristic values. If you chose the option **both** in the presentation section in the preferences popup then you will see two columns for the characteristic name and two columns for the value. The left column always contains the language independent, the right column contains the language dependent description, respectively.
  Table 4-2 shows the content of language independent and language dependent formatting. You may input your values in either format.
  In the test case database, the Sybit Model Tester stores language independent values only. This ensures that test cases may be used across languages.

Value Type	Language Independent	Language Dependent
Date Value	TTMMYYYY e.g.: Dec. 15th, 2009: 15121009	MM.TT.YYYY e.g.: Dec. 15th, 2009: 12.15.2009
Numeric Value	No mask and no unit, decimal point is used as decimal separator e.g.: 12345.67	Formatted with mask and unit e.g.: 12,345.67 pc

Currency Value	Unformatted without currency e.g.: 12345.67	Formatted with currency e.g.: 12,345.67 EUR
Character Value	Language independent identifier e.g.: CMF_PLUG	Language dependent description e.g.: Comfort plug
Time Value	SSMMss e.g: 15 hours 26 min 57 sec: 152657	SS:MM:ss e.g: 15 hours 26 min 57 sec: 15:26:57

Table 4-2: Mapping of language independent and language dependent characteristic values.

- **Type column**: In this column – that has no column header – the application informs you about special types of characteristic values. The following two icons may be displayed:
  - **Object characteristic** : All values that are assigned to reading object characteristics carry the icon for object characteristics. Reading object characteristics are used to transport values e.g. from the sales document into the configuration.
    The application marks these characteristics because in a test run, object characteristic values are set in a separate execution steps before anything else is set.
  - **Delete master data default :** In some cases you may not only wish to set values in the input configuration, but also to delete certain default values. This applies particularly, if you want to create a configuration that requires that you delete a value that is set as a master data default. To learn how to mark a value as a deleted master data default, see below in the description of the value help popup for characteristic values.
- **Value help button**: The value help button opens a popup dialogue that helps you to find characteristic values.

A click on the value help button opens a value help popup window, see Fig. 4-19. In the left **Select** column you may choose predefined characteristic values. If the characteristic is extensible, you may add additional, non-predefined values in the last line of the table. Here, each input of a new value needs to be confirmed by pressing the Enter key. Then, the value is moved into the upper part of the table.

The check boxes of those values are active, that are marked as master data default in the column **Del MD Def**. If you check the check box of a value here, the value will be marked with the appropriate icon in the input configuration (see above). If a value is marked as a deletion of a master data default this has the same effect as if – e.g. in the transaction CU50 – you delete a system set default value manually.

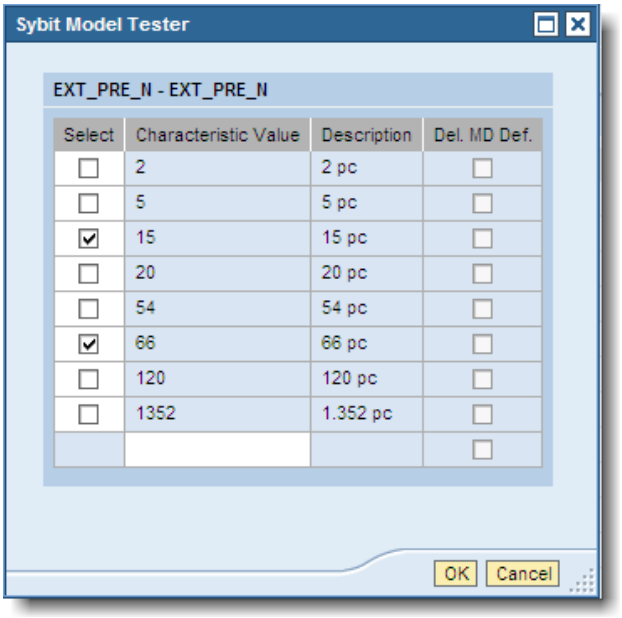


Fig. 4-19: Value help window for characteristic values.

4.7.4 Expected Result

The expected result contains the following tabs. One for expected messages, one for the expected configuration, one for the expected bill of materials, and one for expected prices, respectively.

4.7.5 Expected Messages

In the tab for **expected messages** (Fig. 4-17, on the right) you can specify messages for the conflicts and warnings that you expect the LO-VC to generate upon your input configuration. E.g., if you want to test if a characteristic is marked required, you just leave it empty in the input. If the expected messages contains the warning CSTIC_INCOMPLETE, then the test run succeeds if (and only if) the characteristic is required.
With the search input field you can conduct a full text search on the table. You may choose among all possible messages by means of the drop down box. The messages are listed in Appendix A.

SMT 2.0

If a message can be assigned unambiguously to a characteristic, you may jump directly to the characteristic in the input configuration. To do this, just click on the row selector of the message. The row selector of a row is the button at the beginning of each line that enables you to select the row. If you click here, the input configuration table will scroll so that the according characteristic is displayed in the first line.

4.7.6 Expected Configuration

In the tab for the **expected configuration** (Fig. 4-20) you specify the characteristic values that you expect the system to set based on your input configuration. For your convenience, the values you set in your input configuration are copied automatically into your expected result. Toggling between full view and quick view also affects the input configuration. For a description of these options, see Section 4.7.3.

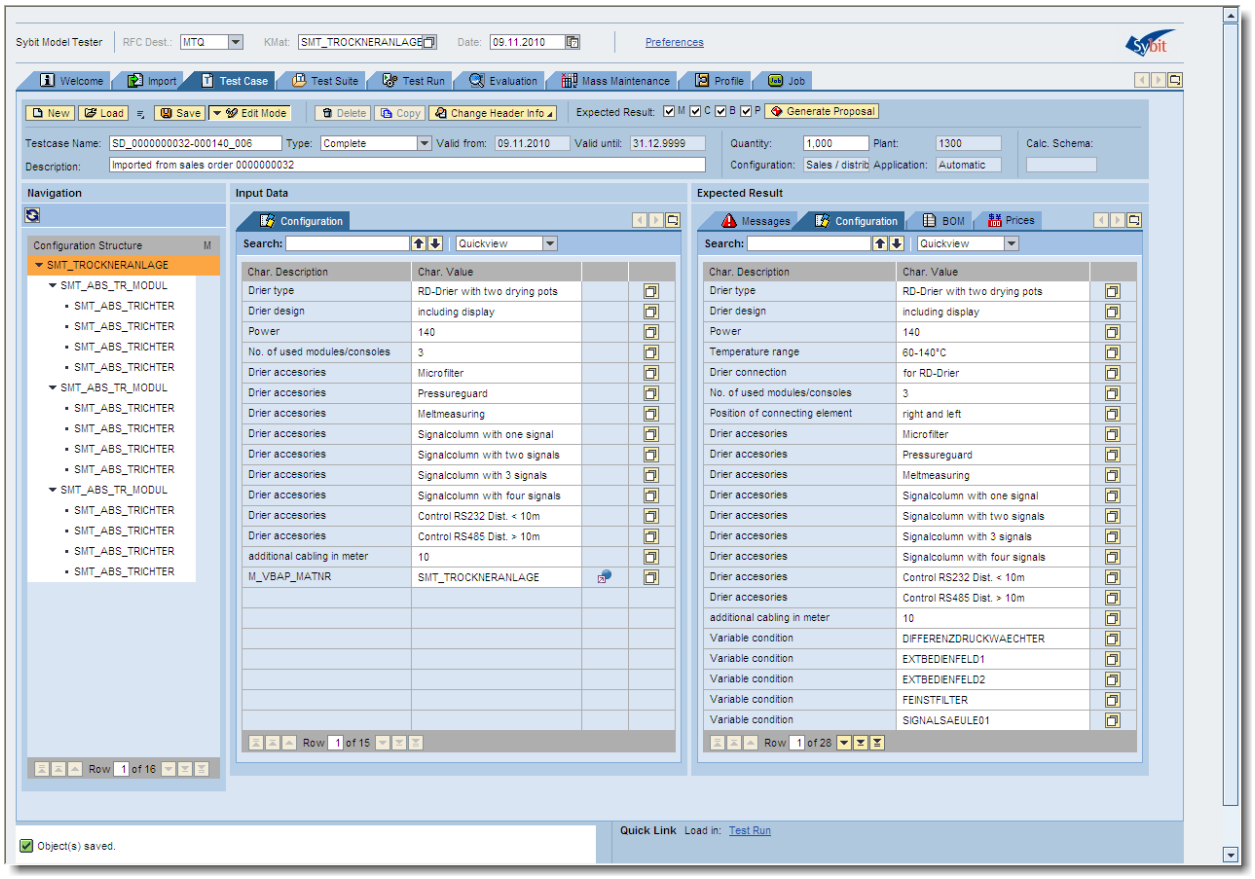


Fig. 4-20: Test case tab: Configuration area of the expected result.

4.7.7 Expected Bill of Materials (BOM)

The tab for the **expected bill of materials (BOM)** contains the result of the BOM explosion (Fig. 4-21). Again, you can use the search field to conduct full text search on the table.

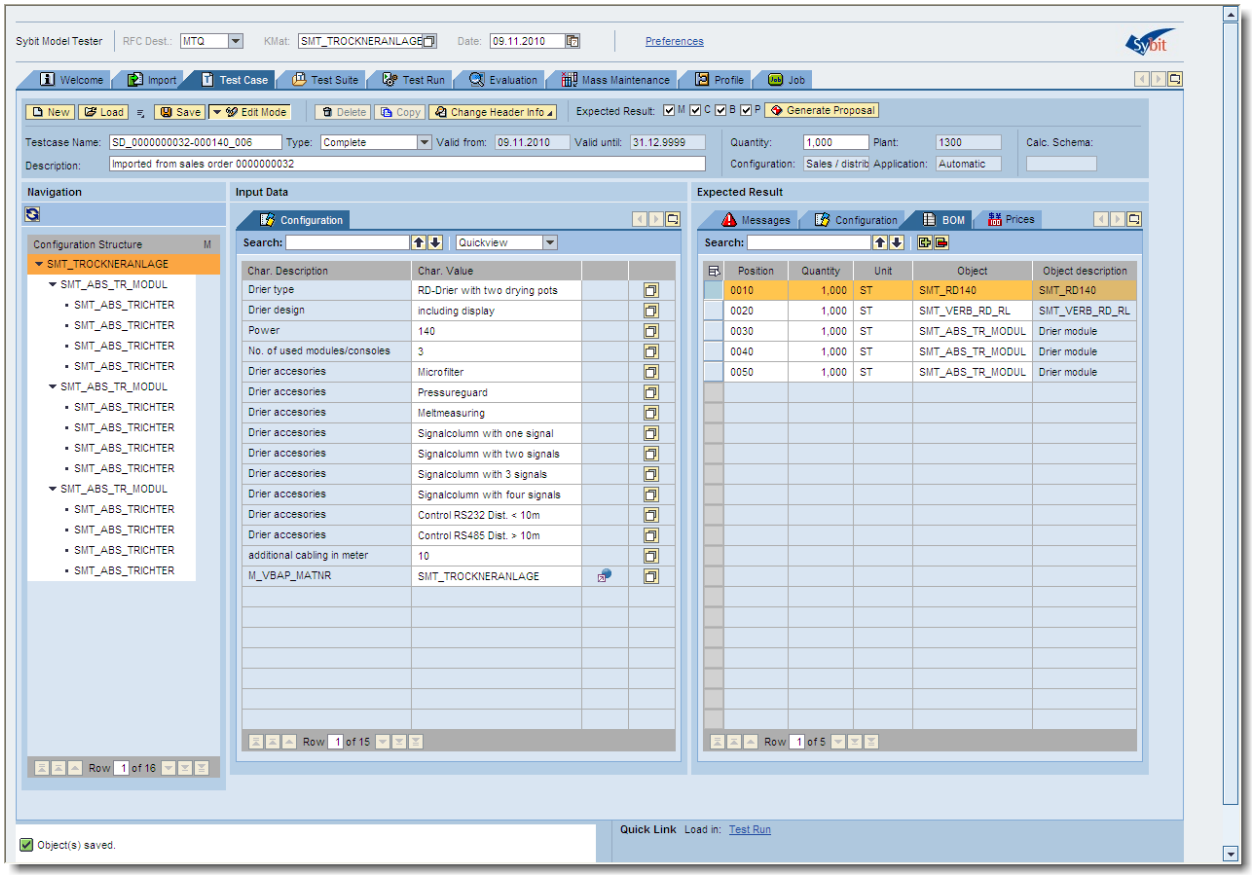


Fig. 4-21: Test case tab: BOM area of the expected result.

4.7.8 Expected Prices

SMT 2.0

In the tab for expected prices you specify all condition records relevant for the variant conditions of your configuration. Only those condition records are tested that are relevant to the calculation schema specified in the header of the test case.

To understand the testing of pricing better, we first have a quick look at the Model Tester for relevant aspects of pricing in SAP (see Fig. 4-22). There, a calculation scheme contains a number of condition types. Each condition type specifies a sequence (10, 20,…) of accesses, the so-called access sequence.

An access is basically a table that has condition records as rows. This table has at least the following columns:

- **A**: Amount
- **C**: Rate unit (Currency or percentage)
- **M**: Condition unit
- **p**: Condition pricing unit

Additionally a sequence may define a number of keys (i.e. columns of the table) that are used to determine a specific condition record and its price. A typical key column is e.g. the variant condition which is used to bind prices to specific configurations. In an access sequence, accesses are always ordered from specific to general, i.e. with increasing access numbers, accesses get less specific. Hence, the last access contains the smallest number of key columns.
In the example displayed in Fig. 4-22 access 10 of condition type 1 contains four columns for sales organization (SalesOrg), distribution channel (DistChan), material (Material), and variant condition (Variant). In contrast, access 20 only contains columns for material and variant condition.

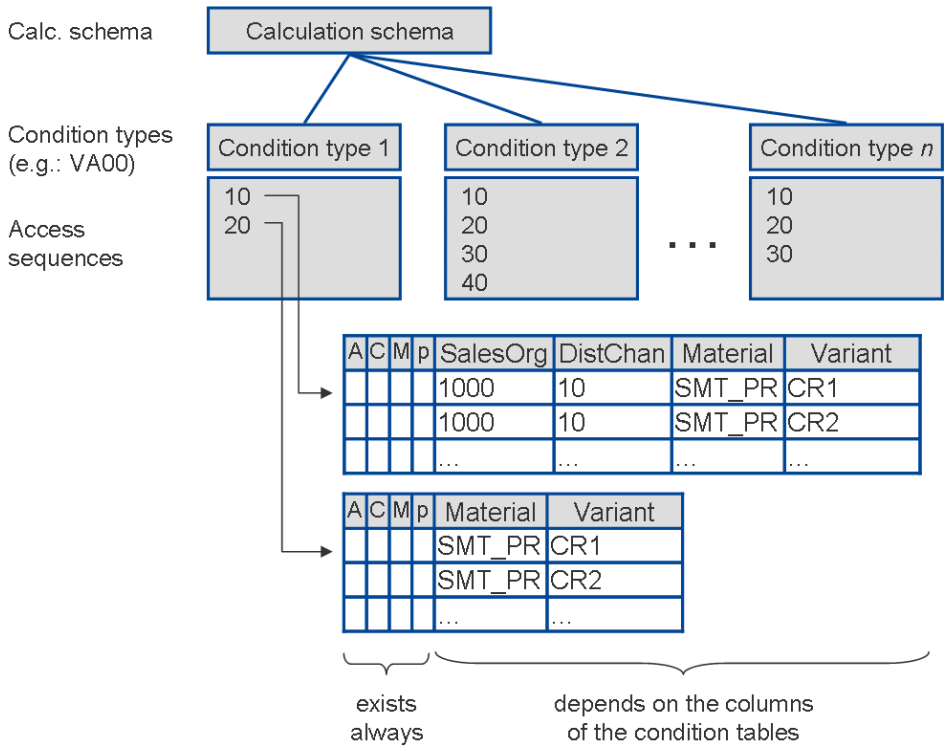
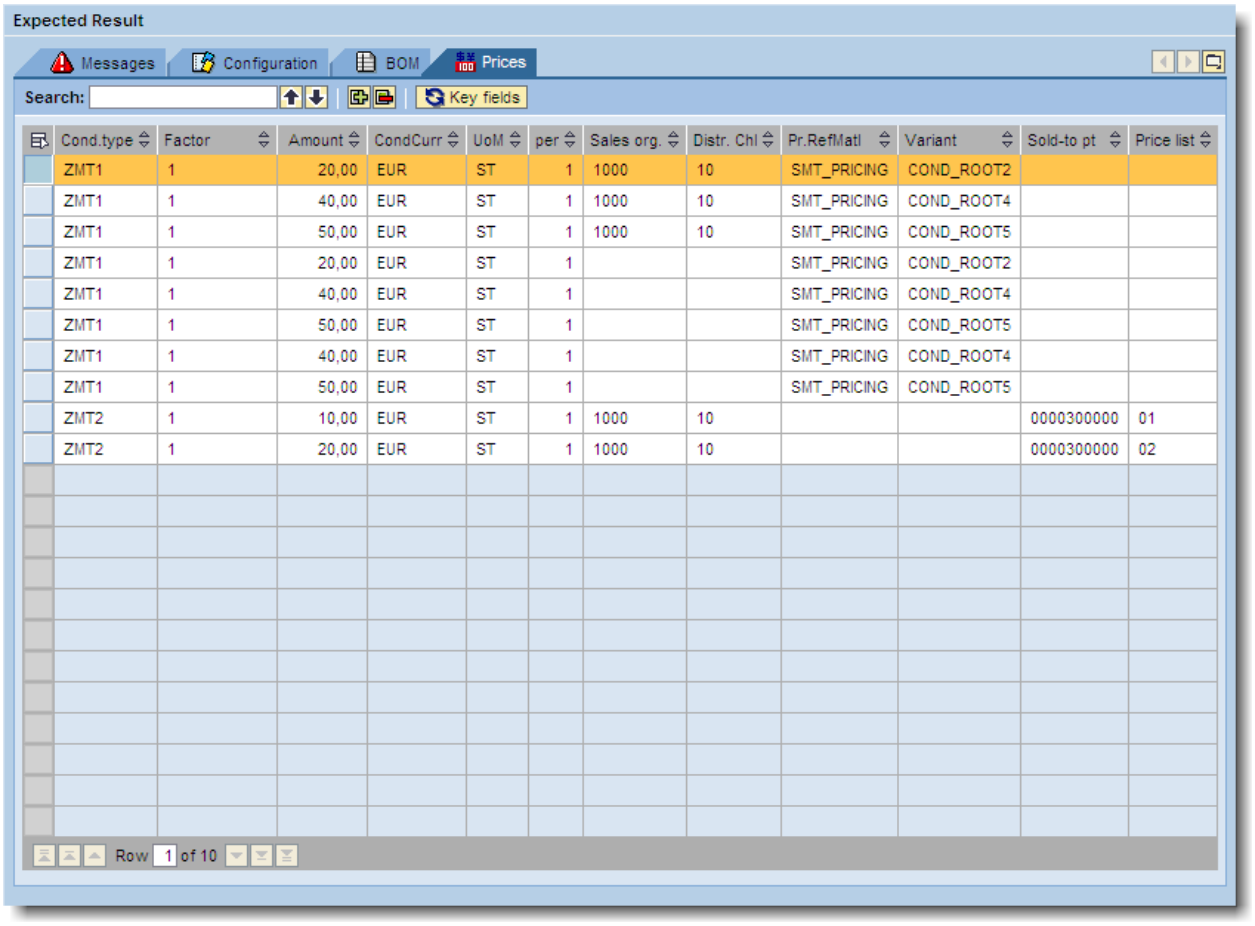


Fig. 4-22: Excerpt from structures relevant to pricing.

The Sybit Model Tester checks all condition records that are relevant to the calculation scheme specified in the test case. Here, relevance is always determined by variant conditions that result from characteristic values in the input configuration of the test case. For all condition types of the calculation scheme, all access sequences and, hence, all accesses are evaluated. This results in a list of all relevant condition records.

This list of condition records is displayed in one single table in the expected result and may also be edited there. The table columns are as follows (see Fig. 4-23):

- **Condition type**: Name of the condition type.
- **Factor**: During model evaluation, the VC calculates a factor for each condition record. This factor is displayed here.
- **Amount, CondCurr, UoM, per:** Static columns for condition records, see above.
- **Dynamic columns:** Since each access may define its own key columns, we see all key columns of all accesses here. Columns that are not needed in an access stay empty for all condition records of this access.



Cond.type	Factor	Amount	CondCurr	UoM	per	Sales org.	Distr. Chl	Pr.RefMatl	Variant	Sold-to pt	Price list
ZMT1	1	20,00	EUR	ST	1	1000	10	SMT_PRICING	COND_ROOT2		
ZMT1	1	40,00	EUR	ST	1	1000	10	SMT_PRICING	COND_ROOT4		
ZMT1	1	50,00	EUR	ST	1	1000	10	SMT_PRICING	COND_ROOT5		
ZMT1	1	20,00	EUR	ST	1			SMT_PRICING	COND_ROOT2		
ZMT1	1	40,00	EUR	ST	1			SMT_PRICING	COND_ROOT4		
ZMT1	1	50,00	EUR	ST	1			SMT_PRICING	COND_ROOT5		
ZMT1	1	40,00	EUR	ST	1			SMT_PRICING	COND_ROOT4		
ZMT1	1	50,00	EUR	ST	1			SMT_PRICING	COND_ROOT5		
ZMT2	1	10,00	EUR	ST	1	1000	10			0000300000	01
ZMT2	1	20,00	EUR	ST	1	1000	10			0000300000	02

Fig. 4-23: Test case tab: Prices area of the expected result

The following controls are available:

- **Add row / delete row** : For manual maintenance of pricing information, you may use these buttons to add / delete rows in the table.

- **Refresh key fields**: This button analyses the calculation schema specified in the header of the test case and determines all key columns of all relevant accesses. New columns are added as empty columns in the table. Columns that cease to exist are deleted.

  Caution: If columns are deleted, all rows are deleted that have a non-empty value in the deleted column.

The Sybit Model Tester always tests if the following two rules are met, independently of the table in the expected prices tab

- **At least once:** Each variant condition must occur in a least one condition record.
- **Last access:** If a variant condition is contained in a condition record of an access of an access sequence, then it must also be contained in the last access of the same access sequence. This rule makes sense as access sequences go from specific to general.

If one of those rules is violated, an according error message occurs in the message tab, see Appendix A.3.

4.8 Test Suite

Test suites are used to organize test cases and help you to execute, evaluate, maintain, and re-find them. The test suite tab contains the central tool to view and edit test suites (Fig. 4-24). On the left hand side you can select test cases by means of the selection component discussed in Section 0. Subsequently, you insert them into your current working suite by clicking the **Insert** button.

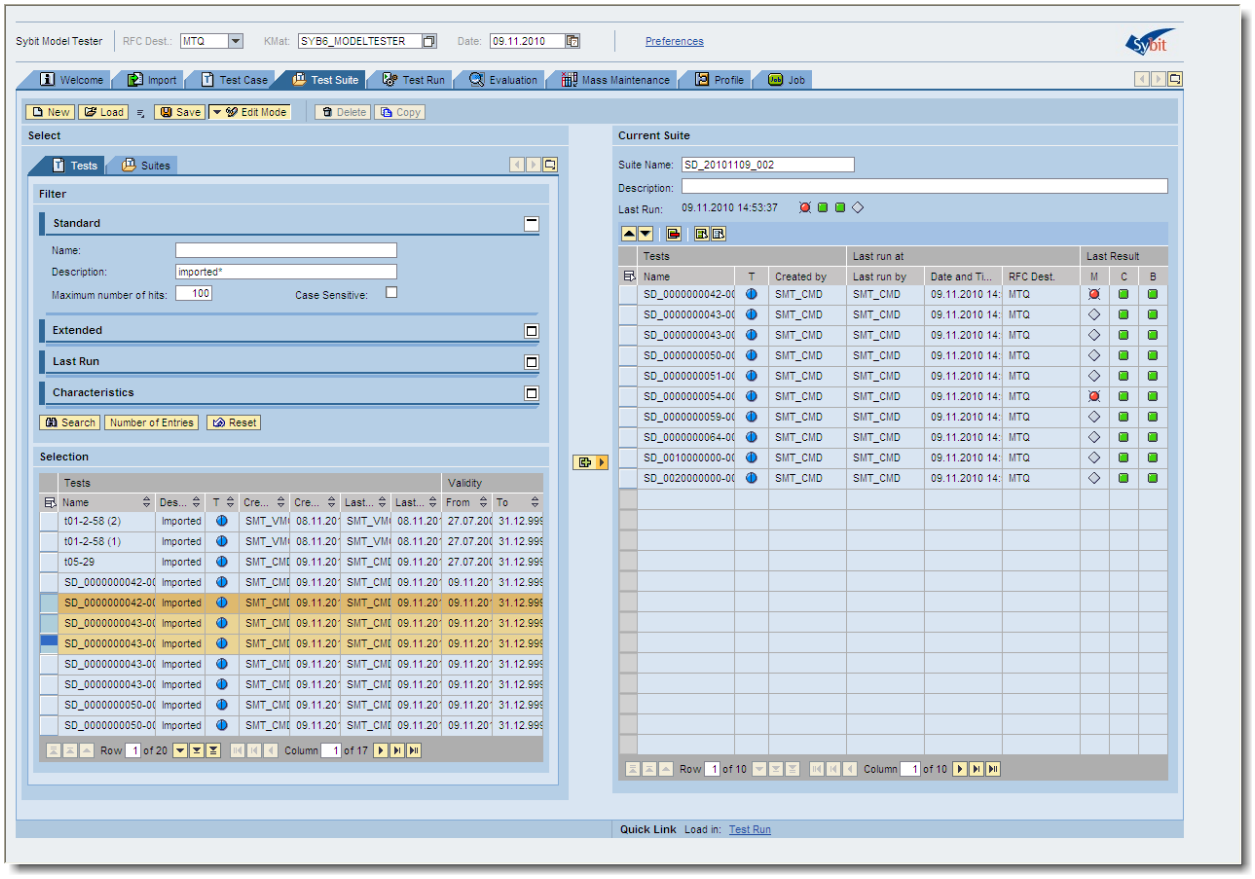


Fig. 4-24: Test suite tab.

The following controls are available:

- **Toolbar**: The buttons **New / Load / Save / Edit Mode / Delete / Copy** fulfill exactly the same functions as discussed in the above section on the test case tab.
- **Selection area**: See Section 4.4.
- **Insert Button**: The button inserts the selected test cases from the left hand side to the current working suite.
- **Current Suite**:
  - **Name / Description / Last Run**: These fields give general information about the suite. The rules for test suite names match those for test case names, see Section 4.7.1. The last run fields denote the date and time when the test suite was last executed.

- **Move Up / Move Down**: The buttons move the selected test cases upwards / downwards within the suite.
- **Remove**: Removes the selected test cases from the suite. The test cases are not removed from the system.
- **Select All / Deselect All**: Selects / deselects all test cases within the test suite.

4.9 Test Run

The test run tab contains the tool to execute test cases and suites (Fig. 4-25).

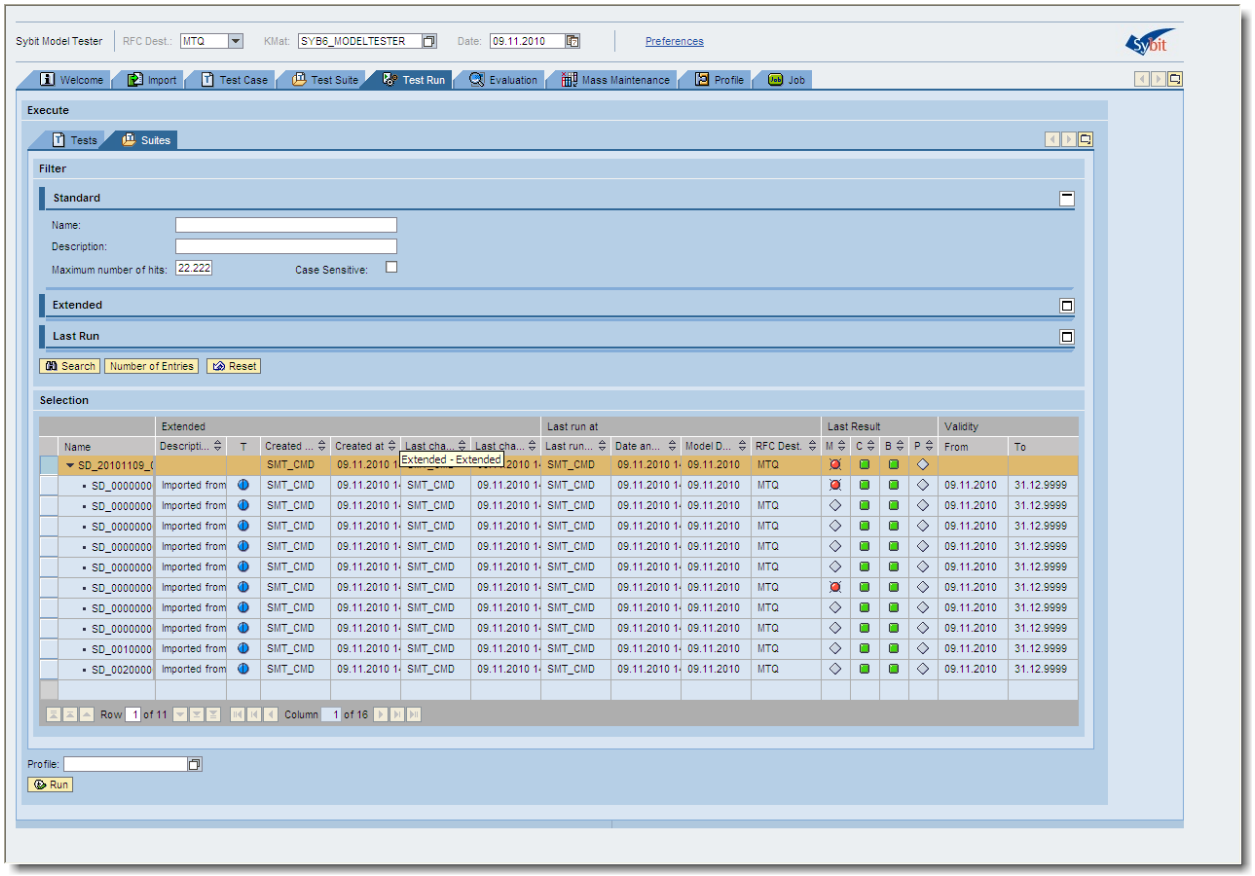


Fig. 4-25: Test run tab.

The following controls are available:

- **Selection Area**: The selection area contains the above described common interface component for test case and test suite selection, see Section 0. Before you run a test or a test suite, you must select it here.
- **Profile:** For each test run you may specify a profile that effects the evaluation of test cases contained by the run, see Section 4.12. In particular a profile may specify that certain errors are ignored on evaluation, i.e. they don't result in a red light. If you don't specify any profile, all information in the test case are considered relevant. Any error during evaluation will result in failure of the respecting test case.
- **Run button**: The test case / test suite selected above is executed in the LO-VC.
  If your scenario contains multiple instances of the LO-VC (this is the case if you have a drop down box labeled "RFC Dest." in the header) the test is run against the LO-VC in the currently displayed system.
  The execution date, i.e. the logical model date where the run is executed is taken from the date in the header.

4.10 Evaluation

The evaluation tab contains the tool to evaluate test runs, see Fig. 4-26.

The Sybit Model Tester application is designed to remember all test runs including the input, the expected result, and the factual result. Hence, for each test case and test suite all test runs can be evaluated at any time. That way, you can retrace how your configuration model behaved at the execution time of each test run.

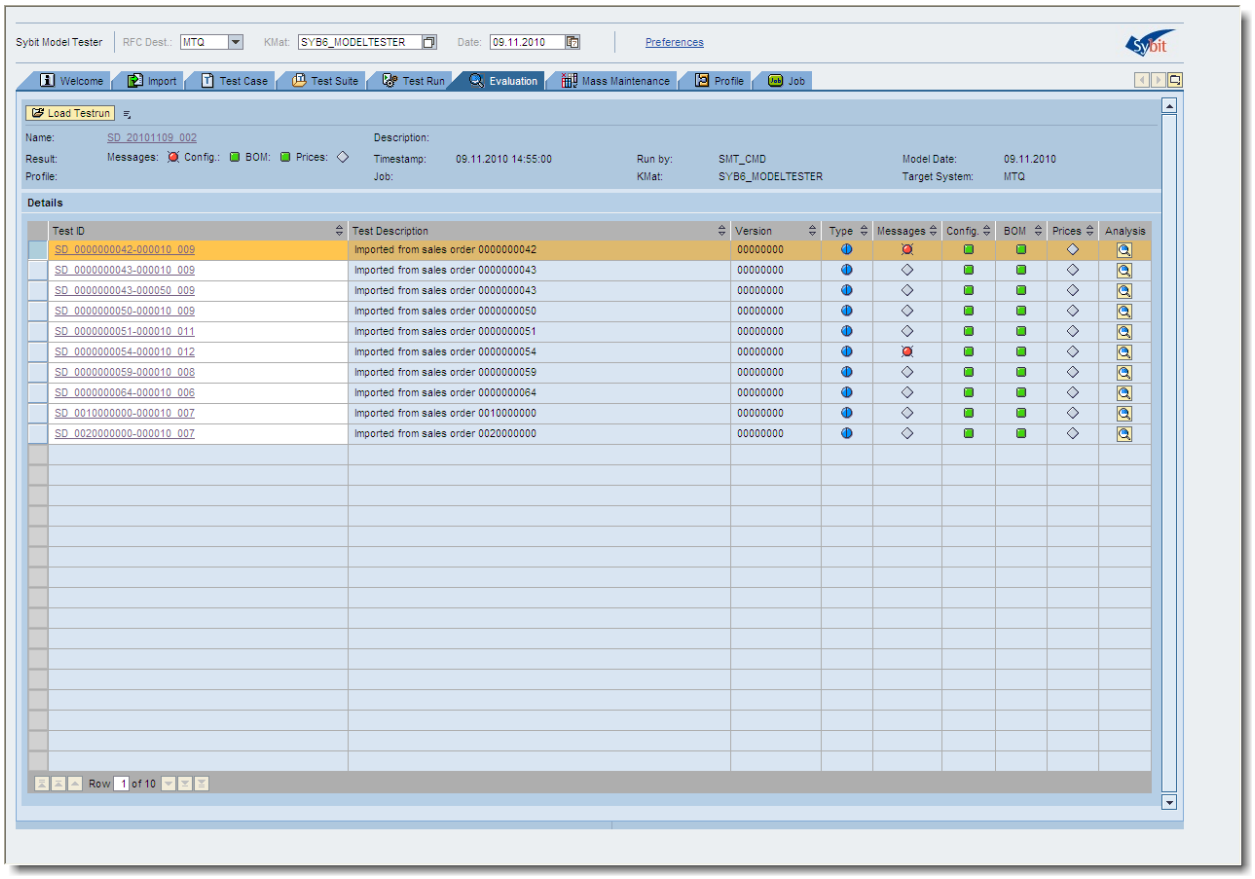


Fig. 4-26: Evaluation tab.

The following controls are available:

- **Load Test Run**: Loads the run of a test case or a test suite for evaluation. The button opens a popup dialogue that works similar to the common selection component for test cases and test suites as described in Section 4.4.
- **Details table**: If you loaded a test suite, the details table contains the list of test cases within the suite. To give you a quick overview, each test case directly shows its state for each of its areas (messages, configuration, BOM, and prices). Only if all test cases in a test suite are green in one area, this area is also green for the test suite.
  If you loaded a test run with one test case only, the table contains one single line only.
- **Analyze** button: To analyze a test case in detail you may press the analyze button which opens the analysis popup window, see Fig. 4-27.

4.10.1 Analysis Window

The analysis window is divided into an area for information and navigation on the left and the main analysis area on the right.

The upper area on the left contains the test run area which displays the name of the test case or test suite that has been executed in this test run. If the test run contains a test suite, you may navigate within the test suite using the buttons **Previous Test case** and **Next Test case**.

The test case area below contains a tool bar with the following button:

- **Adapt Result**: With a click on this button you overwrite the expected result of the current test case with the factual result determined by the test run. Please keep in mind that icons don't become green instantly. To achieve this, you have to re-run the test.
  If your test run has been executed with a profile, the factual result contains the displayed information only. Only this displayed information is copied to the expected result. If you run into this scenario, you will be informed by a popup dialogue that you must confirm.

In the toolbar below, an information area shows you important header data of the test case under analysis. Below that, the following controls are available:

- **Previous Error**: If you analyze a multi-level test case, this will set the focus on the level with the previous error in the navigation tree.
- **Next Error**: If you analyze a multi-level test case, this will set the focus on the level with the next error in the navigation tree.
- **Raise / Show**: You may switch between two view modes via the dropdown box:
  - **Raise errors in navigation tree**: Red icons are increased from the bottom up. If the configuration area of a lower level has a red icon, then all configuration areas of all superior levels will also have a red icon. However, the configuration area itself may have a green icon, which is then displayed in the tab on the right side.
  - **Show errors on respective level only**: Red icons are only shown at the level where the problem occurs. If your tree is not fully expanded and you only see green icons, the test case may have failed anyway which results in red icon in the summary.

In the **navigation area** below you can navigate test cases for multi-level KMats similar to the test case tab, see Section 4.7. For each KMat, the navigation tree shows the respective result divided up in messages (**M**), configuration (**C**), bill of materials (**B**), and prices (**P**).

The **analysis area** has tabs for messages, configuration, bill of materials, and prices, respectively. Each tab is split into two halves. On the left there is a table containing the expected result of the test case. The right side contains a table that shows the factual result. Lines that do not match between left and right are marked with bright red color.

4.10.2 Messages Tab

Fig. 4-27 shows the messages tab. On the left of the analysis area you see that the expected result did contain no messages, whereas the factual result (right side) reports some messages.

You may choose among the following view modes via the drop down box:

- **Show conflicts only**: Only those lines are displayed where expected and factual results do not match. Those lines are red.
- **Show relevant messages** (default value): Rows and columns that are marked irrelevant are hidden.
- **Show all messages**: Irrelevant information is also shown and marked with blue background.

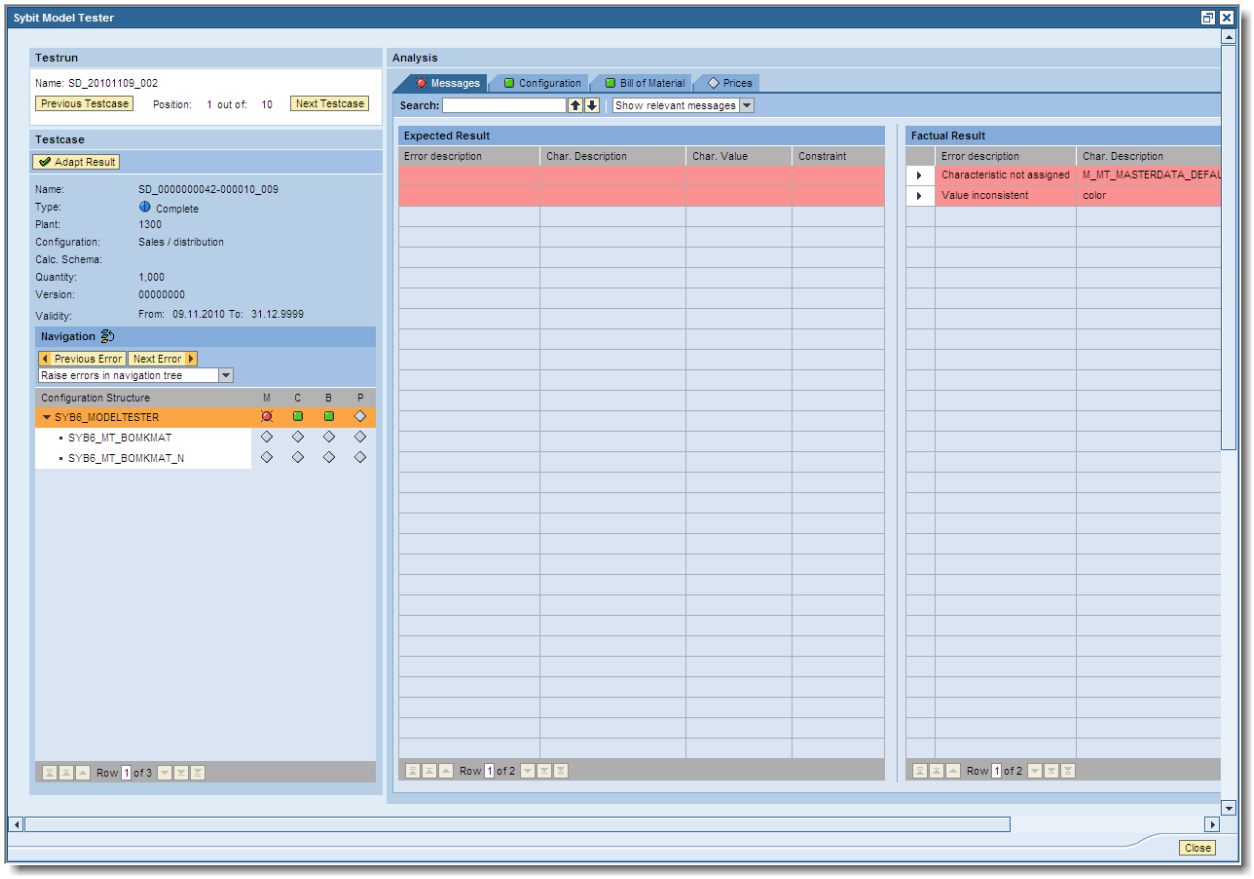


Fig. 4-27: Evaluation tab, analysis popup: Navigation area and messages tab.

4.10.3 Configuration Tab

Fig. 4-28 shows the configuration tab. Here you see only those characteristics that are assigned with an actual value (whereas the test case tab in full view mode also shows characteristics with no values). In the table on the left there is an additional column titled **State**. It tells you if the value has already been part of the input configuration of the test case. A tool tip on the symbol gives you detailed information about the value in the input configuration. You may choose between the following options using the selection area beside the search field:

- **Show conflicts**: Only the lines that contain conflicts (red background) are displayed.
- **Show relevant cstics** (default value): All characteristics that are marked relevant in the profile are shown. Lines with conflicts have red background. If the test run has been executed without any profile all characteristics are shown.
- **Show all cstics**: All characteristics are shown including those that have been marked irrelevant in the profile. Irrelevant characteristics have dark blue background.

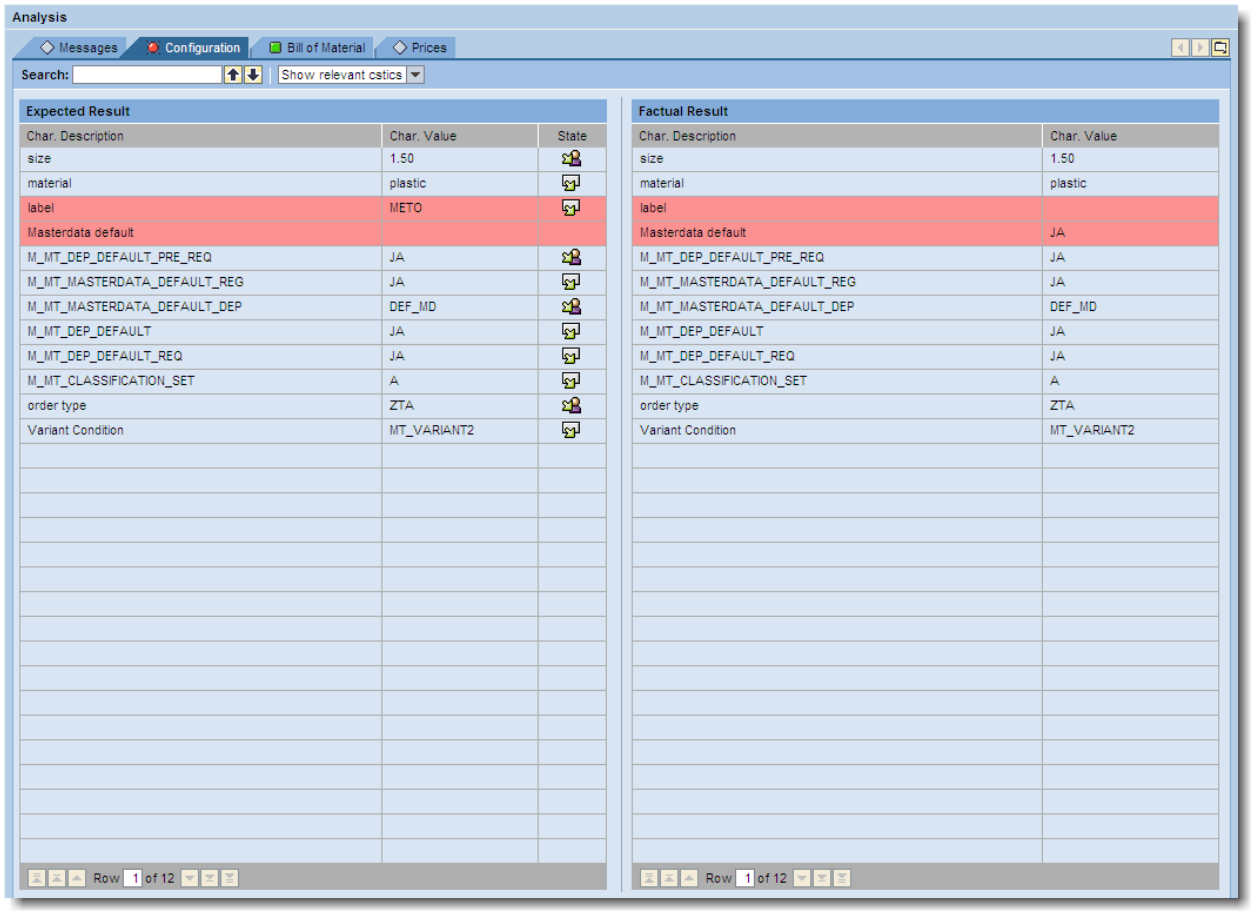


Fig. 4-28: Evaluation tab, analysis popup: Configuration tab.

4.10.4 Bill of Materials Tab

The BOM tab (Fig. 4-29) corresponds in its structure and functionality to the tabs for messages and configuration, see above. You may choose between the following options using the selection area besides the search field:

- **Show conflicts**: Only the BOM lines that contain conflicts (red background) are displayed.
- **Show relevant columns**: All BOM columns that are marked relevant in the profile are displayed. Conflicting lines have red background. If the test run has been executed without any profile all columns are shown.
- **Show all columns**: All BOM columns are shown including those that have been marked irrelevant in the profile. Irrelevant columns have dark blue background.

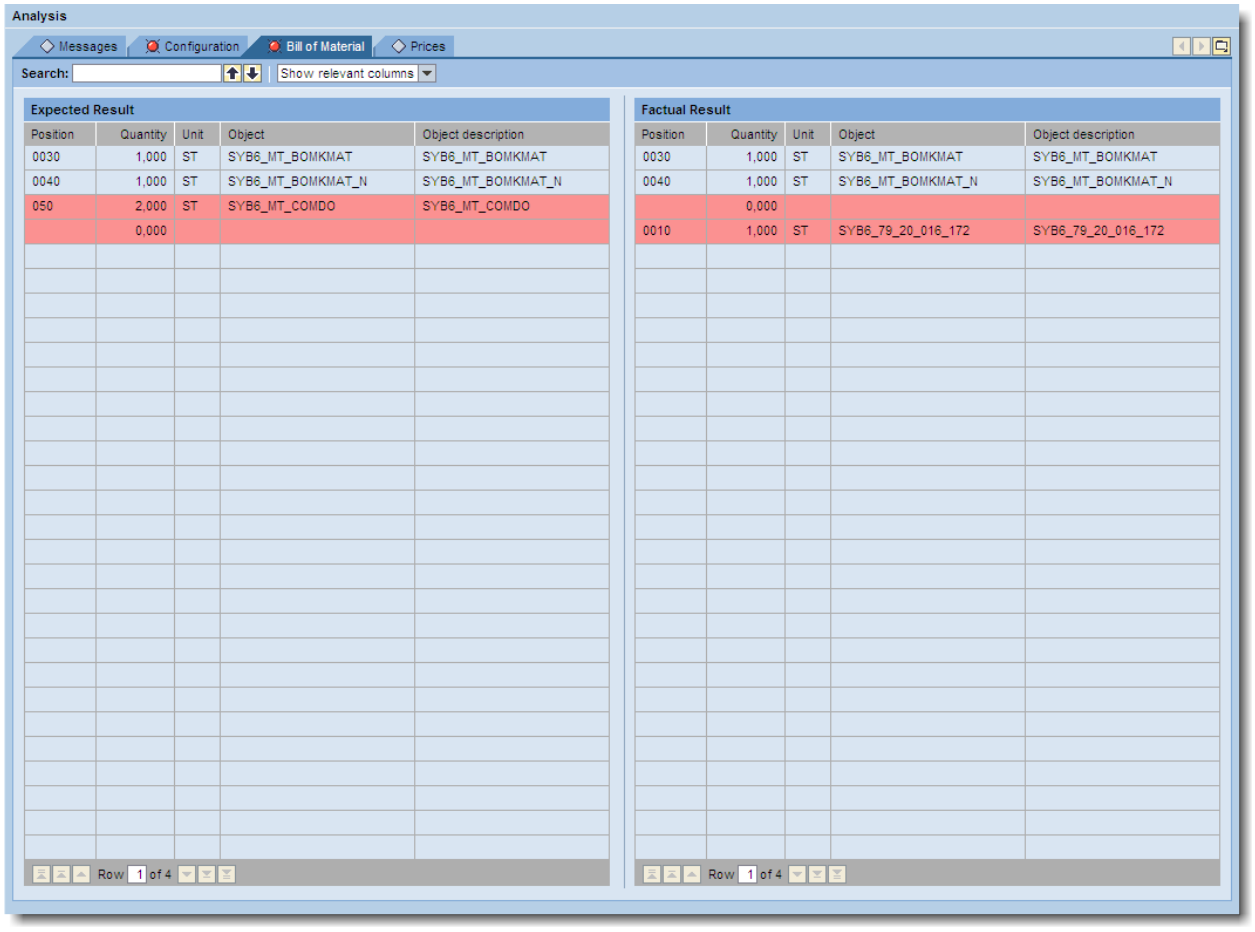


Fig. 4-29: Evaluation tab, analysis popup: BOM tab.

4.10.5 Prices Tab

The prices tab (Fig. 4-30) complies with the above described tabs concerning structure and functionality. You may choose among the following display modes using the drop down box besides the search field:

- **Show conflicts**: Only the condition records that contain conflicts (red background) are displayed.
- **Show relevant columns**: All columns that are marked relevant in the profile are displayed. Conflicting lines have red background. If the test run has been executed without any profile all columns are shown.
- **Show all columns**: All columns are shown including those that have been marked irrelevant in the profile. Irrelevant columns have dark blue background

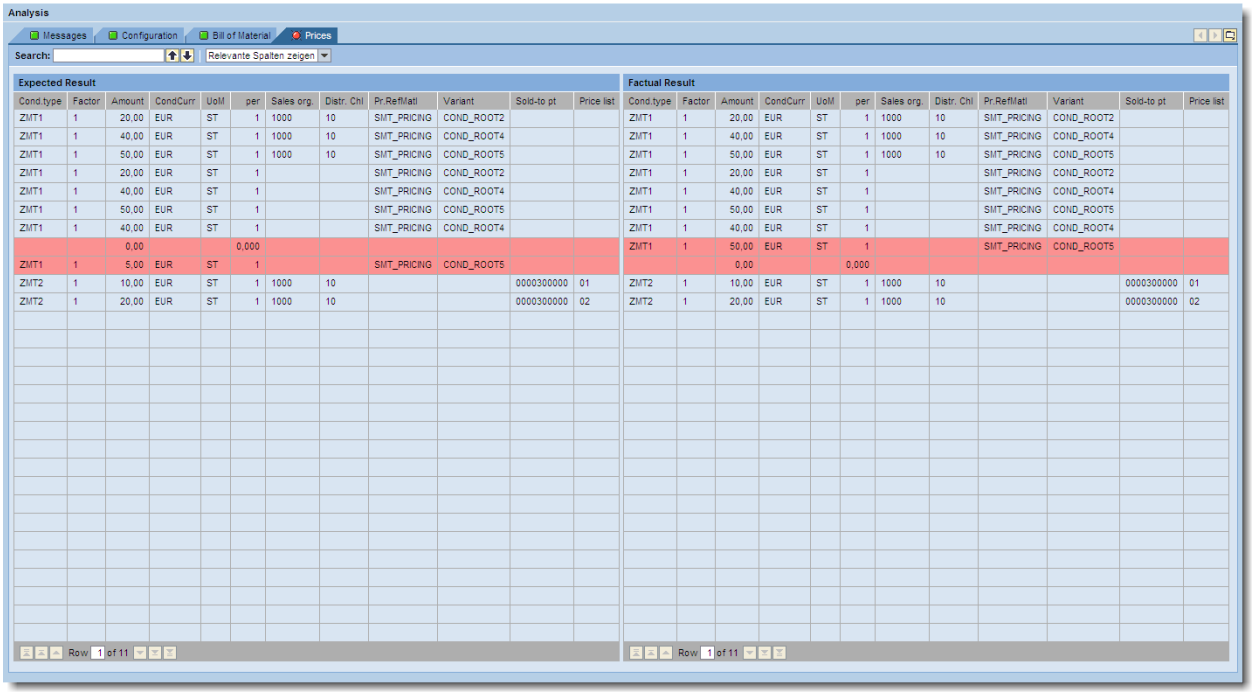

Analysis
Messages
Configuration
Bill of Material
Prices
Search:
Relevante Spalten zeigen
Expected Result
Cond.type Factor Amount CondCurr UoM per Sales org. Distr. Chl Pr.Ref.Matl Variant Sold-to pt Price list
ZMT1 1 20,00 EUR ST 1 1000 10 SMT_PRICING COND_ROOT2
ZMT1 1 40,00 EUR ST 1 1000 10 SMT_PRICING COND_ROOT4
ZMT1 1 50,00 EUR ST 1 1000 10 SMT_PRICING COND_ROOT5
ZMT1 1 20,00 EUR ST 1 SMT_PRICING COND_ROOT2
ZMT1 1 40,00 EUR ST 1 SMT_PRICING COND_ROOT4
ZMT1 1 50,00 EUR ST 1 SMT_PRICING COND_ROOT5
ZMT1 1 40,00 EUR ST 1 SMT_PRICING COND_ROOT4
0,00 0,000
ZMT1 1 5,00 EUR ST 1 SMT_PRICING COND_ROOT5
ZMT2 1 10,00 EUR ST 1 1000 10 0000300000 01
ZMT2 1 20,00 EUR ST 1 1000 10 0000300000 02
Row 1 of 11
Factual Result
Cond.type Factor Amount CondCurr UoM per Sales org. Distr. Chl Pr.Ref.Matl Variant Sold-to pt Price list
ZMT1 1 20,00 EUR ST 1 1000 10 SMT_PRICING COND_ROOT2
ZMT1 1 40,00 EUR ST 1 1000 10 SMT_PRICING COND_ROOT4
ZMT1 1 50,00 EUR ST 1 1000 10 SMT_PRICING COND_ROOT5
ZMT1 1 20,00 EUR ST 1 SMT_PRICING COND_ROOT2
ZMT1 1 40,00 EUR ST 1 SMT_PRICING COND_ROOT4
ZMT1 1 50,00 EUR ST 1 SMT_PRICING COND_ROOT5
ZMT1 1 40,00 EUR ST 1 SMT_PRICING COND_ROOT4
ZMT1 1 50,00 EUR ST 1 SMT_PRICING COND_ROOT5
0,00 0,000
ZMT2 1 10,00 EUR ST 1 1000 10 0000300000 01
ZMT2 1 20,00 EUR ST 1 1000 10 0000300000 02
Row 1 of 11

Fig. 4-30: Evaluation tab, analysis popup: Prices tab

4.11 Mass Maintenance

The mass maintenance tab (see Fig. 4-31) contains a tool that allows you to adapt a large number of test cases at once. The tool is designed primarily to organize configuration model evolvement in the according test set. To realize this, you must always load a set of tests first. Subsequently you may conduct the same operation to all of the tests, such as adding a value for a new required characteristic.

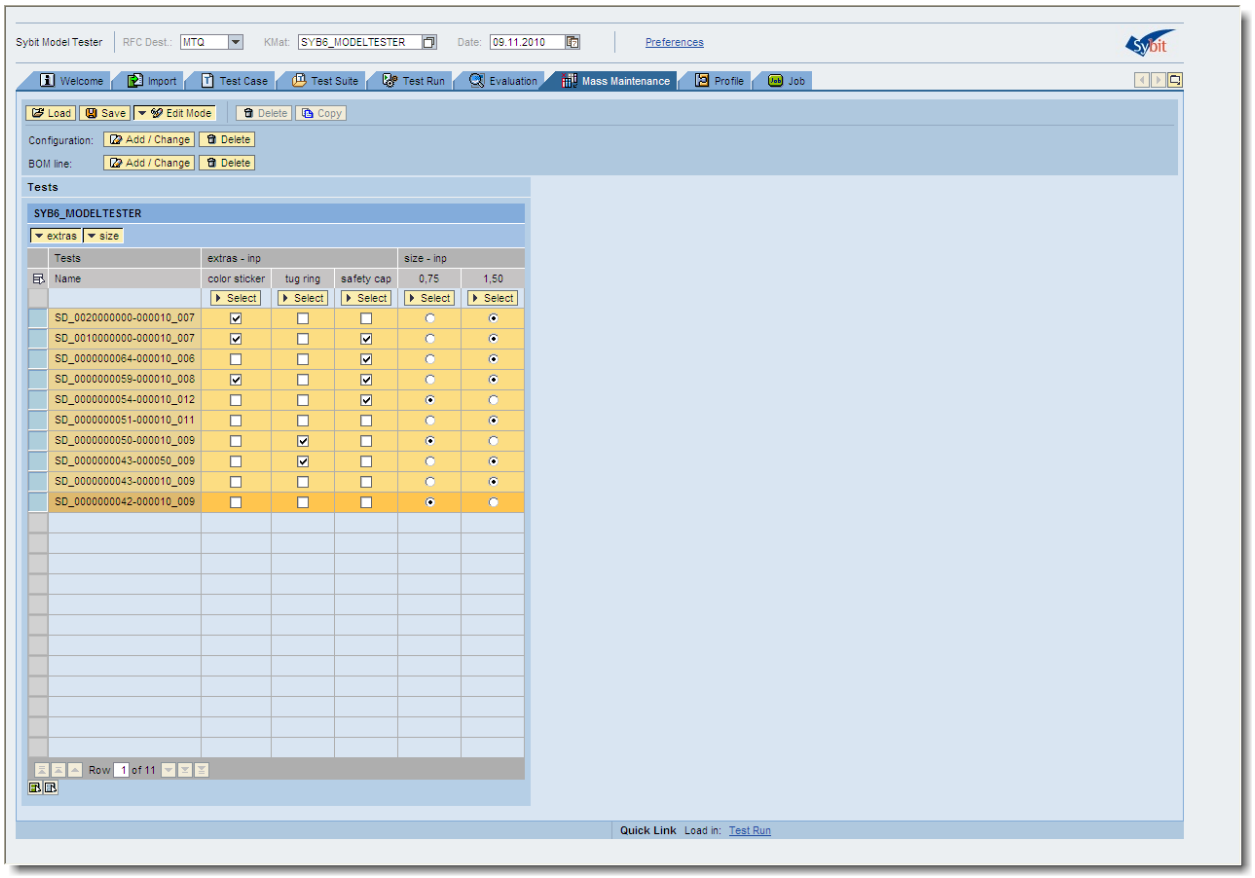


Fig. 4-31: Mass maintenance tab for test cases.

For the maintenance of **multi-level** test cases please note the following: Each test case for a multi-level KMat can be divided into a part for the *main* KMat and into a number of parts, one for each *sub*-KMat. E.g. consider a car as a multi-level KMat that contains a seat KMat. Then the car is the main KMat, the seat is a sub-KMat. If you want to mass maintain a set of multi-level test cases you have to set the KMat that actually changed into the KMat field in the header. E.g., if the configuration model of the seat gets a new required characteristic, you have to enter the seat-KMat into the global KMat field. Then, you can change all test cases that contain the seat as a main KMat or as a sub-KMat in one single action. To clarify on which test cases you are actually working, test cases for sub-KMats inherit the test case name of their main KMat. In the description field of sub-KMats you can find the path where they are located in the actual test case.

Consider the example in Fig. 4-32. Here, we were searching for tests of the sub-KMat SMT_ABS_TR_MODUL and we input the name of a test case that tests a main KMat. As a result,

we get three sub test cases with the identical name. Their description (shown as tool tip) contains the path where they are located inside the test case.

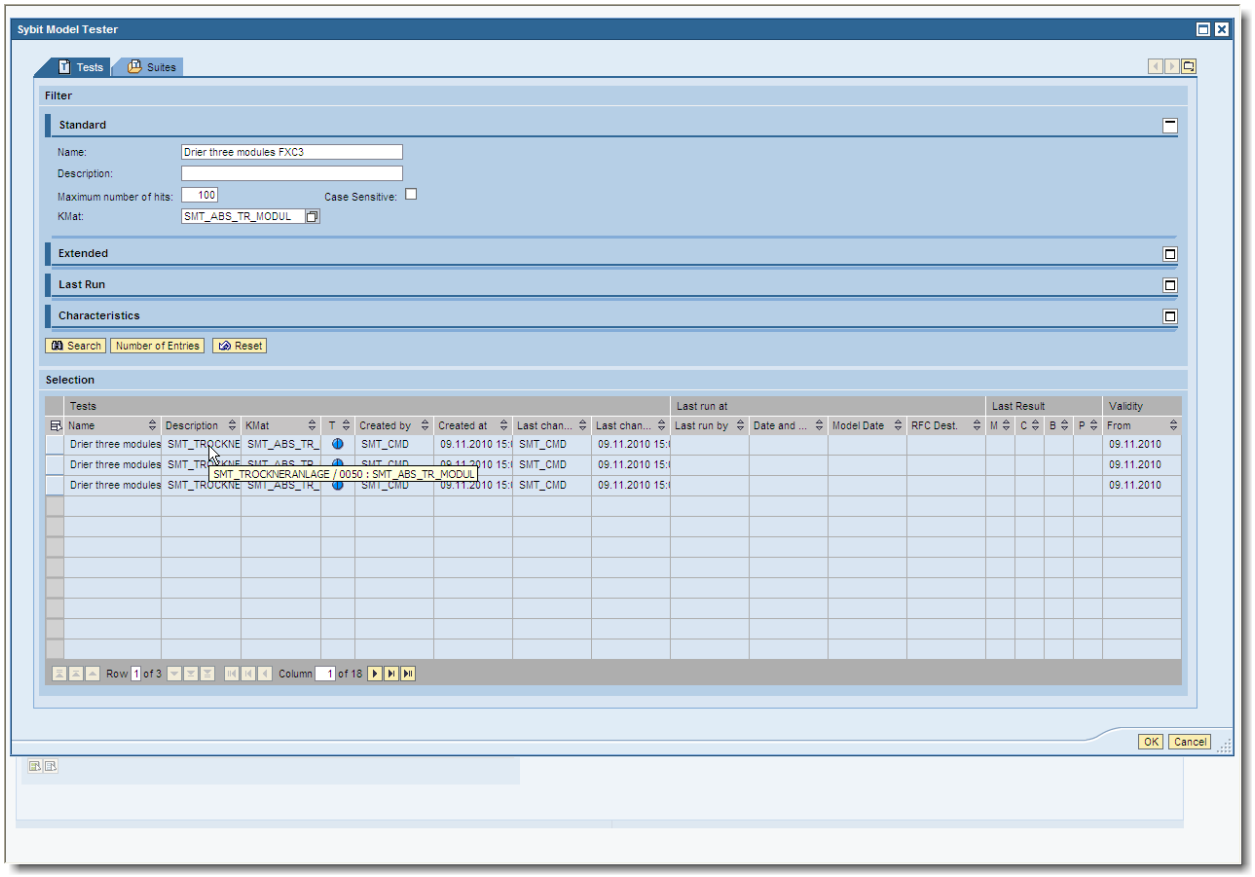


Fig. 4-32: Loading of test for sub-KMats.

In the tool, the following controls are available:

- **Load** / **Save** / **Edit Mode** / **Delete** / **Copy**: These buttons behave just as in the test case tab, see Section 4.7. The only difference is that you load / save / edit / delete / copy multiple test cases and that you can also load tests of sub-KMats as described just above.
- **Configuration – Add / Change**: With this button you can add or change one or multiple characteristic values for all selected test cases. For each characteristic you may choose if the values should be part of the input configuration or of the expected result. In the current version, only the following types or characteristics are supported: Single value characteristics with predefined values, multi-value characteristics with predefined values, and single-valued characteristics without predefined values.
  The characteristics are displayed as a column group in the table below. Each possible value for the characteristic is displayed in a column of its own inside the column group (Fig. 4-31). To set values you have two choices:
  - For characteristics with predefined values, a click on **Select** sets the value for all selected test cases. Likewise, in the case of a text-valued characteristic, the text is assigned to all selected test cases.
  - You can set values on a per-test-case basis by clicking directly into the respective table cell.

- **Configuration – Delete**: All characteristic values of the selected characteristics are deleted in all selected test cases. At the same time, the respective column is hidden in the table.
- **BOM line – Add / Change**: Same functionality as for configuration. For each BOM position there is a column to set its quantity, position, and unit, respectively.
- **BOM line – Delete**: Same functionality as for configuration.

For each characteristic and each BOM line, the table gets a new column group. To show and hide these groups just push the respective toggle buttons on top of the table.

4.12 Profile

The profile tab is used to create and maintain profiles. The Sybit Model Tester uses profiles to fine-tune the behavior of test runs and their evaluation, respectively. Currently, the main task of profiles is to mask – i.e. to hide – foreseeable errors in the evaluation.

Example: Depending on the usage scenario, certain system set characteristics might be irrelevant for the evaluation of the test. E.g. there are characteristics that have only a model internal meaning or there are characteristics that have been made dispensable by model evolution but can not be deleted for technical reasons. If you are not interested in their values, it makes no sense to look at them at all. To this end, these characteristics can be marked to be irrelevant in the profile. If in a test run the expected characteristic value does not match the factual characteristic value, the test case is successful if and only if this characteristic has been marked irrelevant. Another scenario might be that you are not interested in testing the BOM explosion at all, or that for your evaluation the matching of BOM position numbers is not relevant.

Similar to the expected and the factual result, the profile is structured in areas: The messages area, the configuration area, the BOM, and the pricing area. Fig. 4-33 shows the profile tab with its messages area.

In contrast to the tabs for test case, test suite, and test run, the profile tab is independent of the KMat in the header area. Each profile can contain any number of KMats. This implies that you only need one profile for multi-level test cases. Another application for many KMats in a profile is that you may specify one single central company specific profile that defines the relevance of all characteristics in all KMats under test.

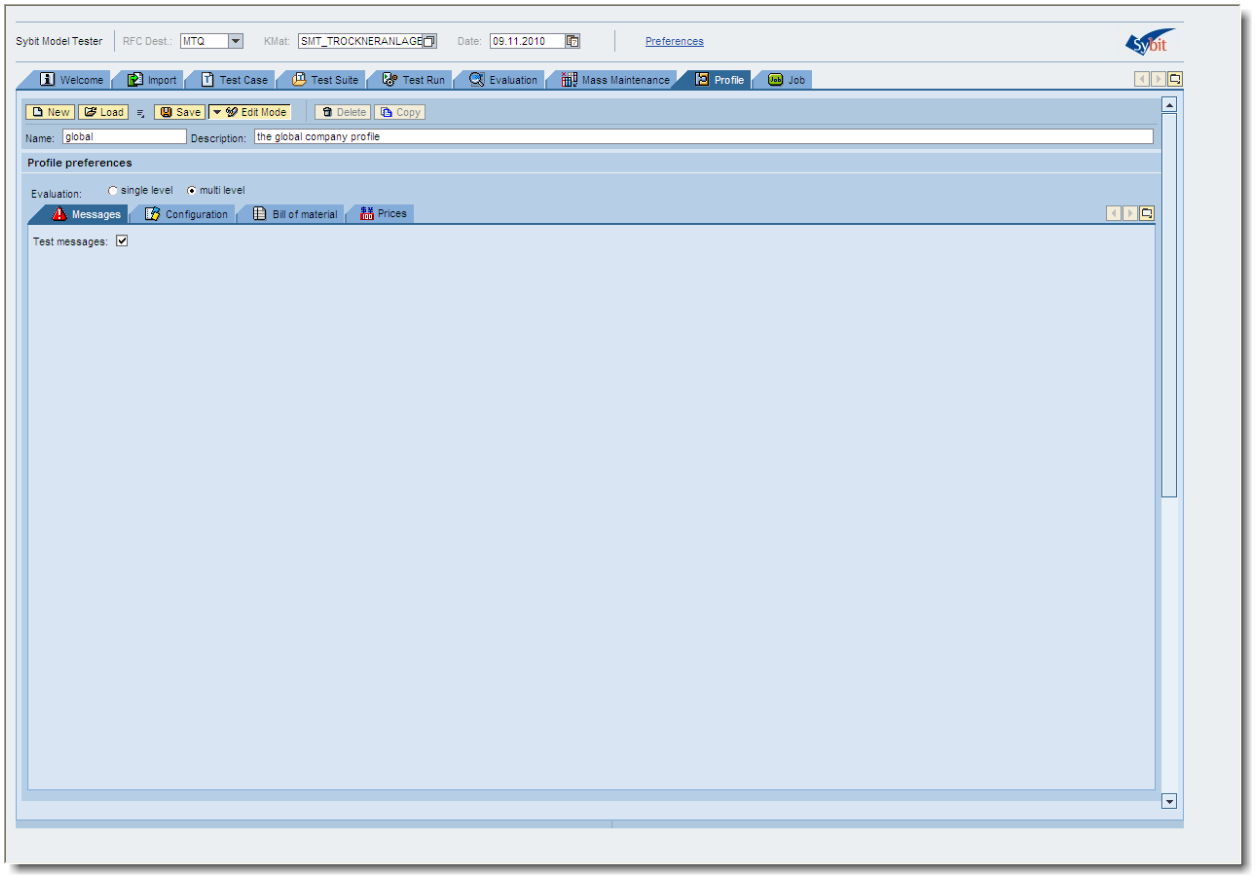


Fig. 4-33: Profile tab: Messages tab.

The following controls are available in the tool:

- **New** / **Load** / **Save** / **Edit Mode** / **Delete** / **Copy**: Similar as in the test case tab, see Section 4.7.
- **Name** / **Description**: Name and description of the Profile. Again similar as in the test case tab, see Section 4.7.

4.12.1 Messages Tab

SMT 2.0

In the current version, the messages tab lets you switch on or off messages, see Fig. 4-33.

4.12.2 Configuration Tab

In the configuration tab you may specify how the Model Tester deals with conflicts in the configuration area, see Fig. 4-34. Here, the main purpose is to hide and disregard predictable conflicts for certain characteristics. For any number of KMats you may specify a list of irrelevant characteristics. If a conflict occurs in an irrelevant characteristic, it will not lead to a failure of the test case.

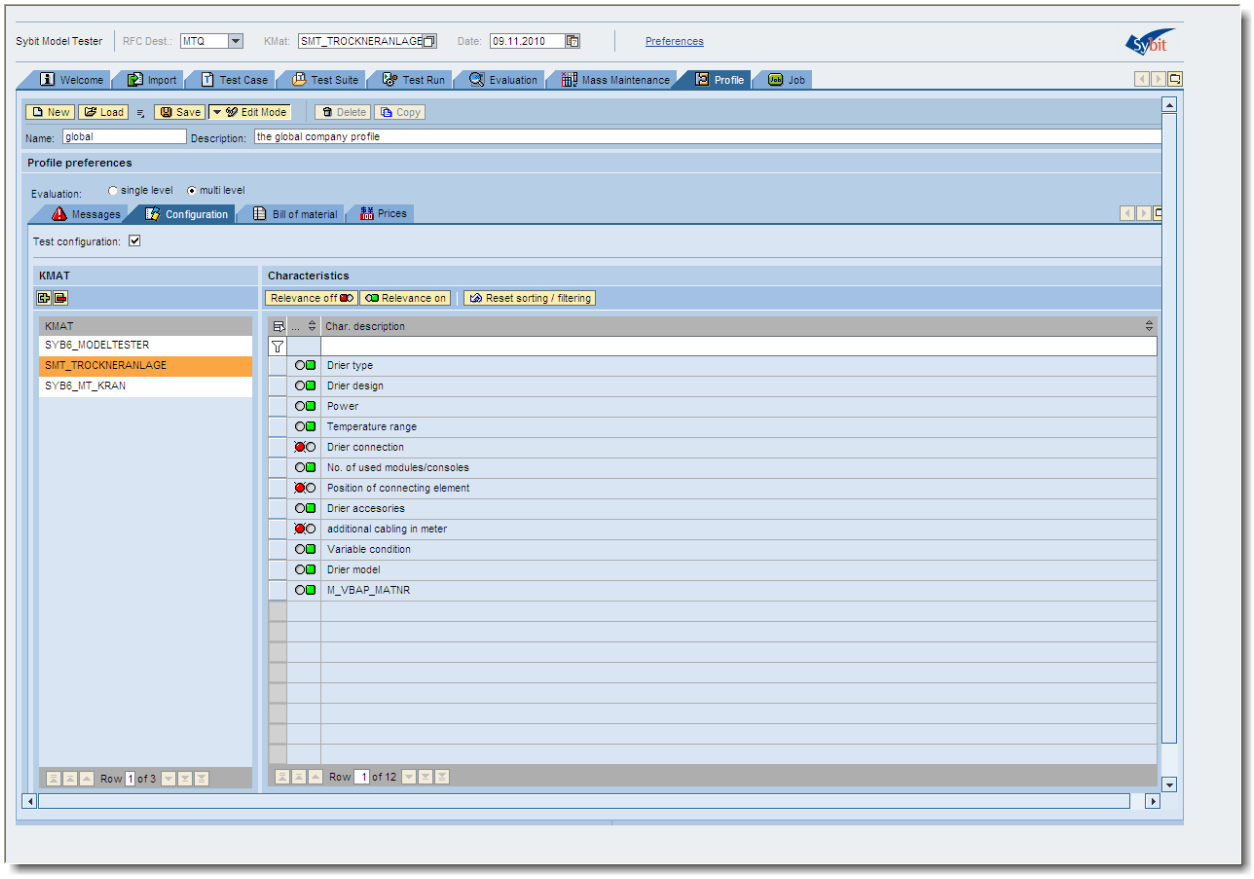


Fig. 4-34: Profie tab: Configuration tab.

The following controls are available:

- **Test configuration**: Use this check box to specify if the configuration area is relevant for you or not *as a whole*. If you uncheck this check box you will never get an error in the configuration area.

KMat table: Shows the list of all KMats for which you may specify irrelevant characteristics in this profile.

- **Add KMat** - Button: Opens a search help window where you may choose a KMat to be inserted into the table.
- **Delete KMat** - Button: Deletes the selected KMat from the profile.

Characteristics-Table: Use this table to define the relevance of characteristics contained by the KMat that is selected in the KMat table.

- **Relevance off**: All selected characteristics are marked irrelevant.
- **Relevance on:** All selected characteristics are marked relevant.
- **Reset sorting:** If you sorted the list of characteristics according to their status, this button resets the sorting. I.e. characteristics are displayed according to their class order.
- **Filter line**: In this line you may specify a filter expression to restrict the displayed characteristics to a suitable subset.

4.12.3 Bill of Materials Tab

In the BOM tab you may specify how the Model Tester deals with conflicts in the BOM area, see Fig. 4-35. Here, the main purpose is to hide and disregard predictable conflicts for certain BOM columns. If a conflict occurs in an irrelevant column, it will not lead to a failure of the test case.

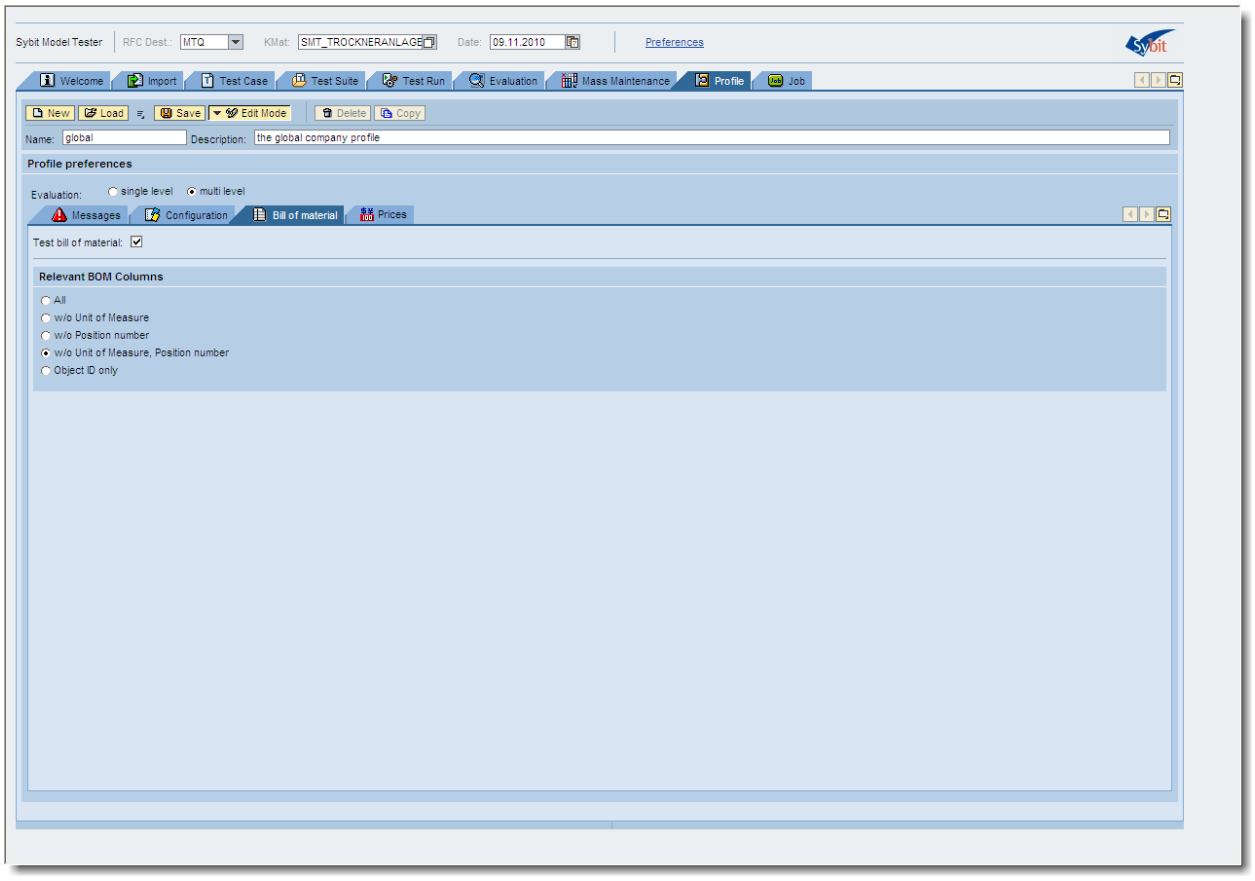


Fig. 4-35: Profile tab: BOM tab.

The following controls are available:

- **Test expected bill of materials**: Use this check box to specify if the BOM area is relevant for you or not *as a whole*. If you uncheck this check box the test run will disregard BOM explosion. Hence the BOM is not tested at all.
- **Relevant BOM columns**: Here you can define which columns of the BOM are relevant. In the Model Tester, the bill of materials contains the following columns: Position, quantity, unit, object ID (i.e. material ID), and language dependent description of the material. Since the Model Tester always relies on language independent IDs, the language dependent description is *never* relevant. The following options are available:
  - **All**: All columns are relevant for evaluation.
  - **W/o unit**: The unit is irrelevant. The position, the quantity, and the object ID are relevant.

- **W/o position number**: The position number is irrelevant. The unit, the quantity, and the object ID are relevant.
- **W/o unit, position number**: The unit and the position number are irrelevant. The quantity and the object ID are relevant.
- **Object ID only**: The unit, the position number, and the quantity are irrelevant. Only the object ID is relevant for evaluation.

4.12.4 Prices Tab

SMT 2.0

In this area you specify how to deal with errors in the prices area of test cases, see Fig. 4-36. You may define which columns of the prices table should be relevant for the success of test run. An error in a column that is marked as irrelevant will not lead to a failure of the test case.

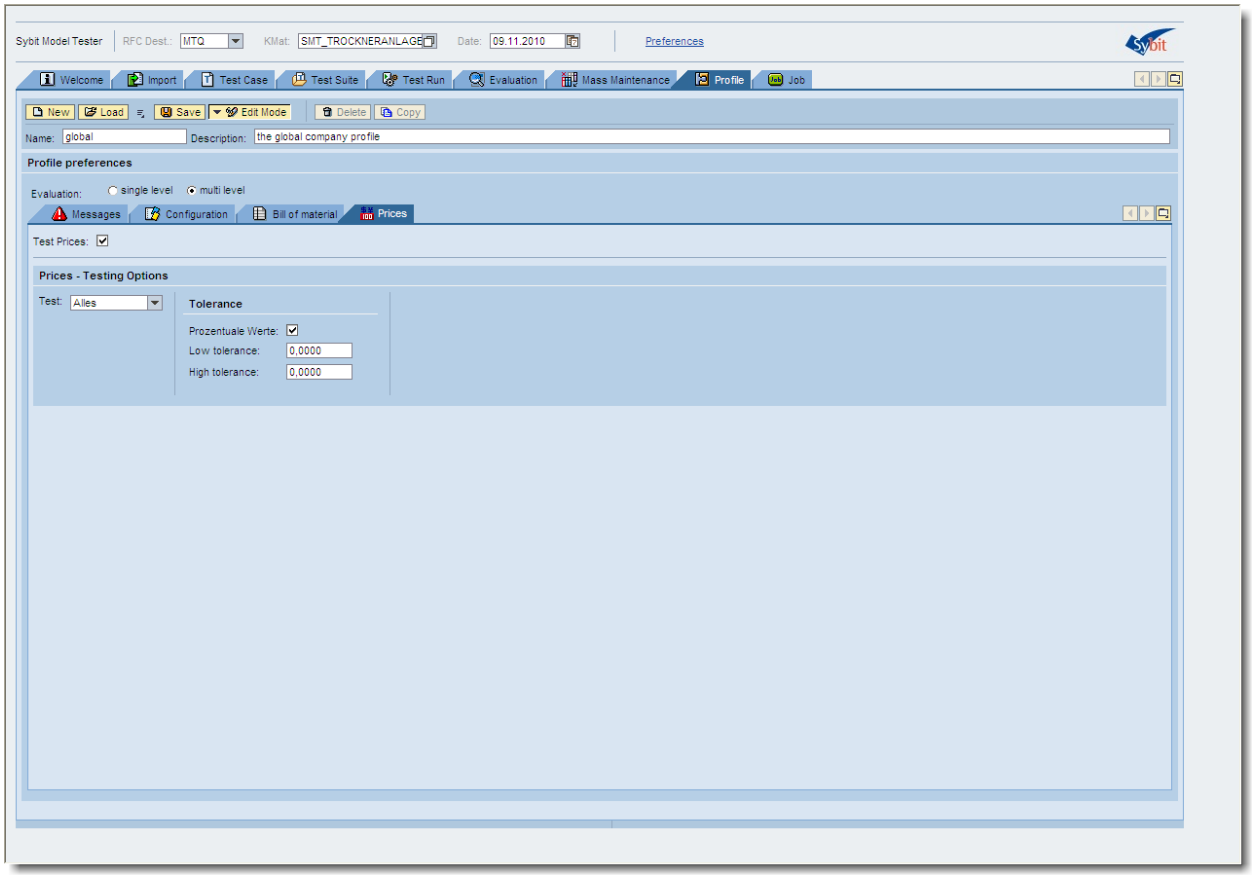


Fig. 4-36: Profile tab: Prices tab.

The following controls are available:

- **Test Prices:** This check box determines if the prices area is tested or not as a whole. If you uncheck this check box no prices will be tested.
- **Test Amount:** In this software version, you may choose among two options:
  - **All:** The analysis of a test run considers all columns to be relevant.

- **W/o amount, currency:** In many scenarios, prices change so quickly that the effort of maintaining prices in test cases would not pay. If you choose this option, the system checks the existence of condition records only.

- **Tolerance:** If you choose to test **All** in the drop down box, you may specify a tolerance area for testing amounts. Using the input fields for **Low tolerance** and **High tolerance**, you may specify how much the amount in the factual result may differ from the amount in the expected result without causing the test to fail. If you choose **percentaged values,** the input values define a tolerance area as percentage of the expected result. Otherwise, the values are treated as absolute amounts.

4.13 Job

The job tab enables you to schedule test runs periodically. The Sybit Model Tester executes these runs at the scheduled times fully automatically. To realize this, you may specify a number of test cases and test suites in the Sybit Model Tester. The scheduling of the job takes place by means of SAP's standard transaction for job scheduling, SM37.

Fig. 4-37 illustrates this relationship. For each job in the Sybit Model Tester – in the following we call this an **SMT-Job**, the system automatically generates a respective job in the SAP system – which we will call **SAP-Job**. SMT-Job and SAP-Job always have identical names.

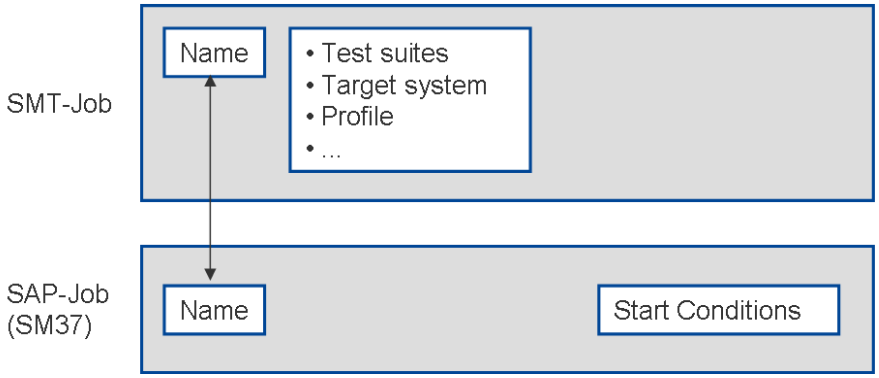


Fig. 4-37: SMT-Job and SAP-Job: Connected via their Names.

The content of the job, i.e. the information which test cases should be executed with which profile and which RFC destination is defined in the SMT-Job. The scheduling information is independently stored in the SAP-Job (transaction SM37).

For performance reasons, on execution of a job each contained test suite is executed as a separate test run. The evaluation of those test runs can be done as for all other test runs in the evaluation tab, see Section 4.10.

Fig. 4-38 shows the job tab.

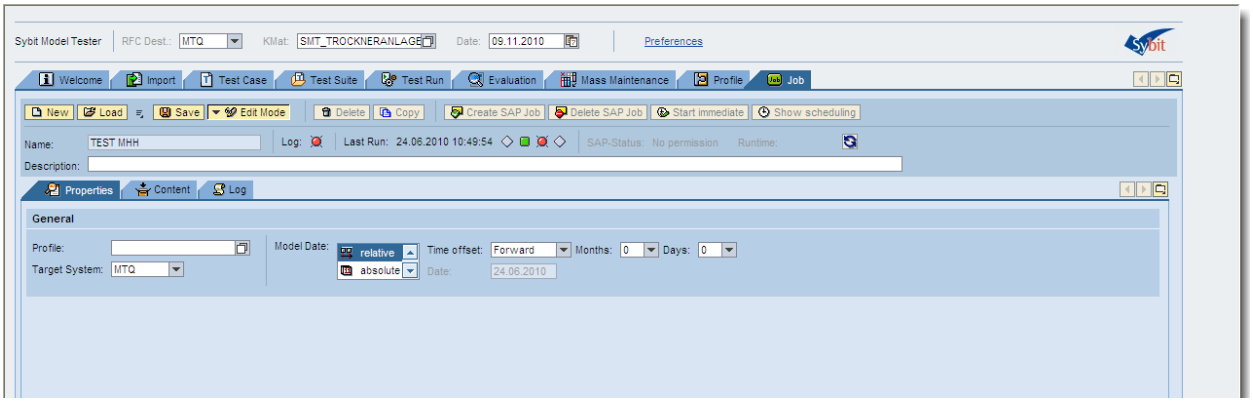


Fig. 4-38: Job tab: Properties tab.

The following controls are available:

- **New / Load / Save / Edit Mode / Delete / Copy**: As in the test case tab (see Section 4.7).

- **Create SAP Job**: This button causes a new SAP-Job to be created in the SAP system. The new SAP-Job has the same name as the SMT-Job. From now on, it can be scheduled via transaction SM37.
- **Delete SAP Job**: This button is the opposite of Create SAP Job. It deletes the appropriate SAP-Job from the SAP system. Data stored in the Sybit Model Tester is not deleted. Only the scheduling information in the SAP system is discarded.
- **Start immediately**: If you press this button, the job you are currently viewing is executed right away. In contrast with a similar button in transaction SM37, this button does not discard the scheduling information of the SAP-Job.
- **Show scheduling**: This button opens a popup dialogue that displays the scheduling information as specified by transaction SM37.
- **Name / Description**: Name and description of the SMT-Job. If an SAP-Job has been created, you can not change the name of the SMT-Job any more. Otherwise you would break the connection between SAP-Job and SMT-Job.

SMT 2.0

- **Log**: Shows the summarized state of the last run of the job. The icon displayed here equals the icon of the run in the subordinate log tab. First, it contains information about the success of all areas of all test cases in all test runs. Second, it indicates if the job as a whole has been executed successfully.
  Consider the following example: A job contains exactly one test suite. On execution time of the job, one test case of this test suite is being edited and – hence – the test case is locked. It can not be run. All other test cases can be run without conflicts. In this scenario, the icon for the job is red whereas the icons for all areas are green. To learn the reason for the red job icon, the user can check the execution log inside the log tab, see below.
- **Last Run**: Time stamp of the last run and icons for the last result of the job for each area.
- **SAP-Status / Runtime / Refresh button**: Here you can see the status of the SAP-Job. If the job is currently running, the runtime field shows the elapsed time. Otherwise, it shows the total runtime of the last run. The refresh button affects the fields for SAP status and runtime only.

4.13.1 Properties Tab

The job tab contains three sub-tabs. Fig. 4-38 shows the properties tab.

The following controls are available:

- **Profile**: Determines the profile with which the job is executed.
- **Target System**: Determines the RFC destination of the target system on which the job is executed.
- **Model Date**: If you execute a job periodically then you have to decide if you want the model date to change in relation with the execution date, or if the job always uses the same (absolute) model date.
  - **Relative**: Here, you may specify a relative date which moves alongside the job execution date into the future. To this end, you may specify a date offset and additionally if the **offset** should be added to (**Forward**) or subtracted from (**Backward**) the execution date. Use the input fields for **Months** and **Days** to specify the offset in the range between 0 and 99.
  - **Absolute**: If you specify an absolute model date, all executions of this job will use the specified date as model date.

4.13.2 Contents Tab

The contents tab is displayed in Fig. 4-39. In its functionality it is very similar to the test suite tab as described in Section 4.8. Please note the following differences:

- **Test suites:** Different from test suites that may contain test cases only, a job may contain multiple test suites. This takes effect in extended search functionality for the table on the right hand side. To use the extended search, press the according button:
- **KMats:** Test suites may contain test cases for one single KMat only. This restriction is not valid for jobs. To this end, on the left hand side, you may choose any KMat.
- **Navigate to Evaluation:** To analyze test runs, the name of each entry in the table on the right carries a quick link that enables instant navigation into the evaluation tab. Additionally, you may also navigate into the test case or test suite tab, respectively. To make the navigation links visible, you just have to click on the name column of the entry.

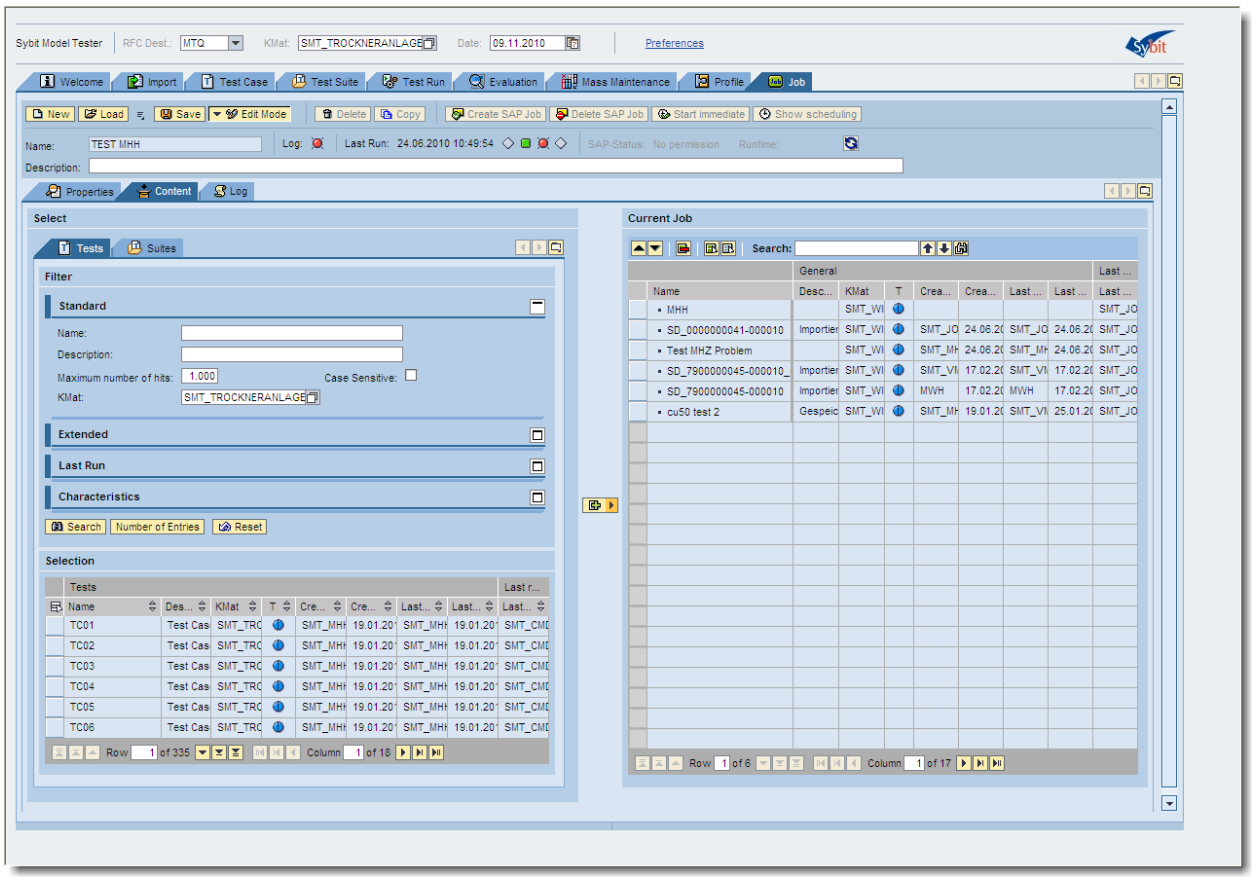


Fig. 4-39: Job tab: Content tab.

4.13.3 Log Tab

The log tab is displayed in Fig. 4-40. It offers a view on information that has been generated automatically during the execution of the job. In the upper **Filter** area you may specify a time span in which you want to search for logs. The **Log** area on the left side contains a **refresh** button that fills the log table below with current information from the system. After pressing refresh, the table contains the logs of all job executions within the specified time span. A click on a time stamp in the table loads the respective log into the table on the right.

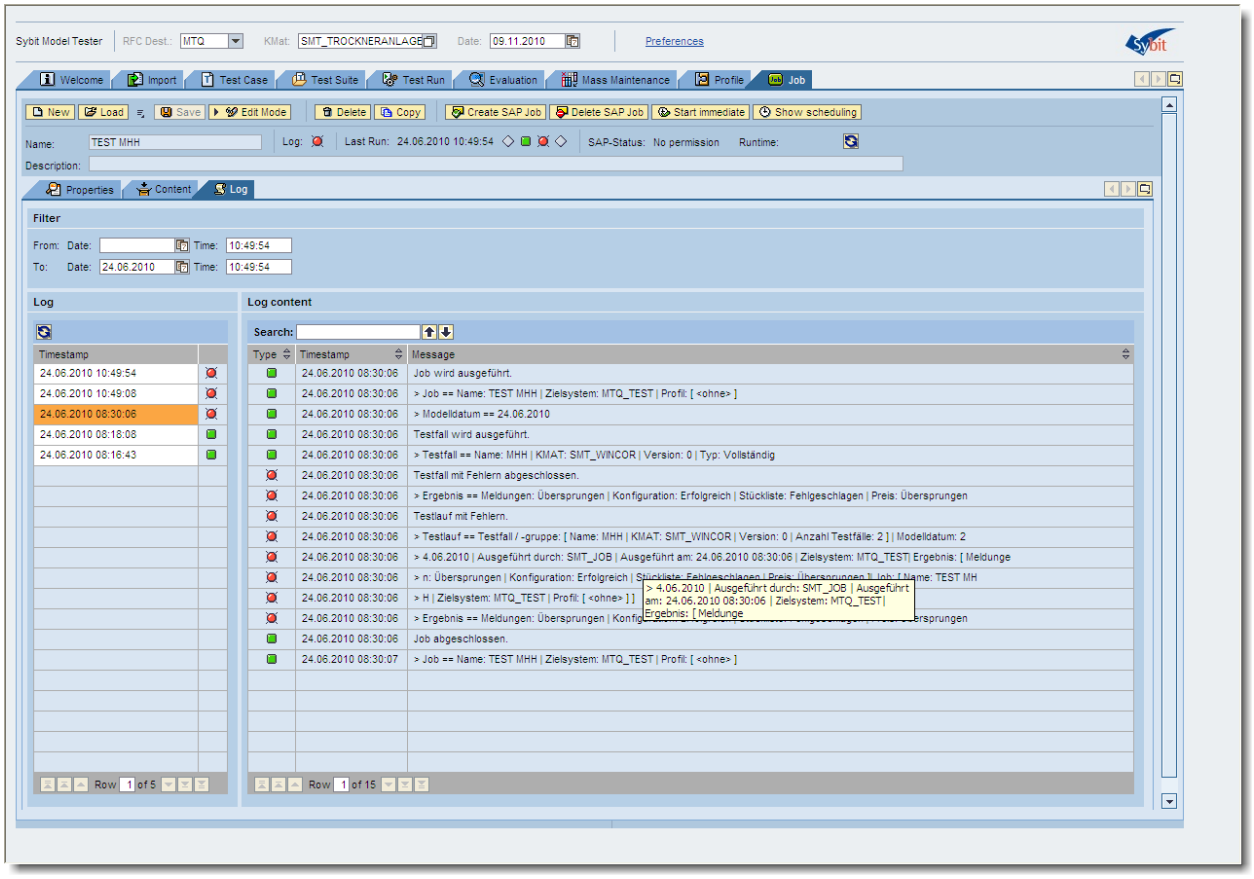

Sybit Model Tester
RFC Dest.: MTQ
KMat: SMT_TROCKNERANLAGE
Date: 09.11.2010
Preferences
Welcome
Import
Test Case
Test Suite
Test Run
Evaluation
Mass Maintenance
Profile
Job
New
Load
Save
Edit Mode
Delete
Copy
Create SAP Job
Delete SAP Job
Start immediate
Show scheduling
Name: TEST MHH
Log:
Last Run: 24.06.2010 10:49:54
SAP-Status: No permission
Runtime:
Description:
Properties
Content
Log
Filter
From: Date: Time: 10:49:54
To: Date: 24.06.2010 Time: 10:49:54
Log
Log content
Search:
Timestamp
24.06.2010 10:49:54
24.06.2010 10:49:08
24.06.2010 08:30:06
24.06.2010 08:18:08
24.06.2010 08:16:43
Row 1 of 5
Type
Timestamp
Message
24.06.2010 08:30:06 Job wird ausgeführt.
24.06.2010 08:30:06 > Job == Name: TEST MHH | Zielsystem: MTQ_TEST | Profil: [<ohne>]
24.06.2010 08:30:06 > Modelldatum == 24.06.2010
24.06.2010 08:30:06 Testfall wird ausgeführt.
24.06.2010 08:30:06 > Testfall == Name: MHH | KMAT: SMT_WINCOR | Version: 0 | Typ: Vollständig
24.06.2010 08:30:06 Testfall mit Fehlern abgeschlossen.
24.06.2010 08:30:06 > Ergebnis == Meldungen: Übersprungen | Konfiguration: Erfolgreich | Stückliste: Fehlgeschlagen | Preis: Übersprungen
24.06.2010 08:30:06 Testlauf mit Fehlern.
24.06.2010 08:30:06 > Testlauf == Testfall / -gruppe: [Name: MHH | KMAT: SMT_WINCOR | Version: 0 | Anzahl Testfälle: 2] | Modelldatum: 2
24.06.2010 08:30:06 > 4.06.2010 | Ausgeführt durch: SMT_JOB | Ausgeführt am: 24.06.2010 08:30:06 | Zielsystem: MTQ_TEST| Ergebnis: [Meldunge
24.06.2010 08:30:06 > n: Übersprungen | Konfiguration: Erfolgreich |
24.06.2010 08:30:06 > H | Zielsystem: MTQ_TEST | Profil: [<ohne>]]
24.06.2010 08:30:06 > Ergebnis == Meldungen: Übersprungen |
24.06.2010 08:30:06 Job abgeschlossen.
24.06.2010 08:30:07 > Job == Name: TEST MHH | Zielsystem: MTQ_TEST | Profil: [<ohne>]
Row 1 of 15

Fig. 4-40: Job tab: Log tab.

4.14 Saving Test Cases from CU50

As an optional extension to the Sybit Model Tester, there is a plugin for transaction CU50 that allows you to save test cases in the Sybit Model Tester. To find out if this plugin is installed in your system, please contact your system administration. In the Sybit Model Tester standard software, the transaction CU50 is not modified. In contrast, we ship a transaction Y_CU50 that contains the additional saving functionality. Apart from the saving functionality, there is no difference between CU50 and Y_CU50.

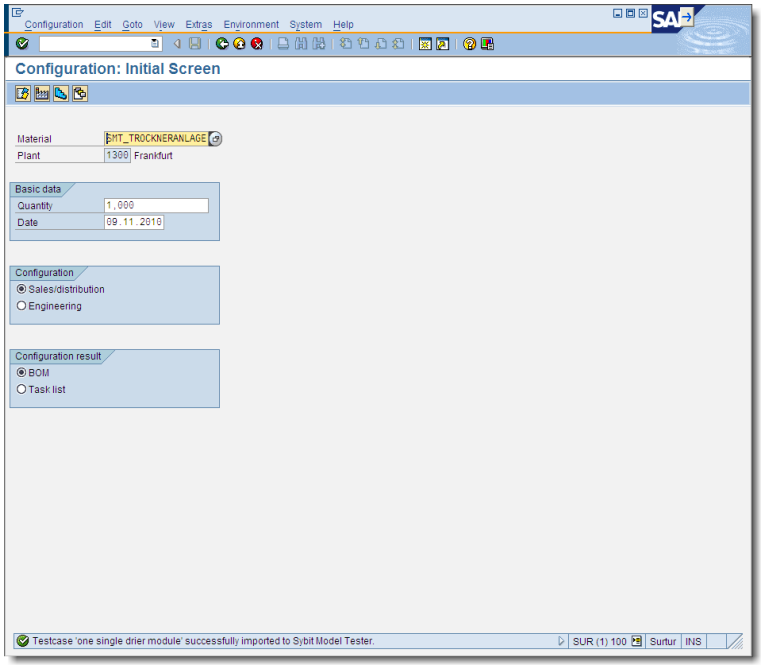


Fig. 4-41: Start screen of Y_CU50.

To trigger the save function you just have to click on the save button in the menu bar. In the standard transaction CU50 this button is disabled. If you enter a complete configuration and press save, CU50 is left and a popup dialogue appears that prompts you to enter a test case name.

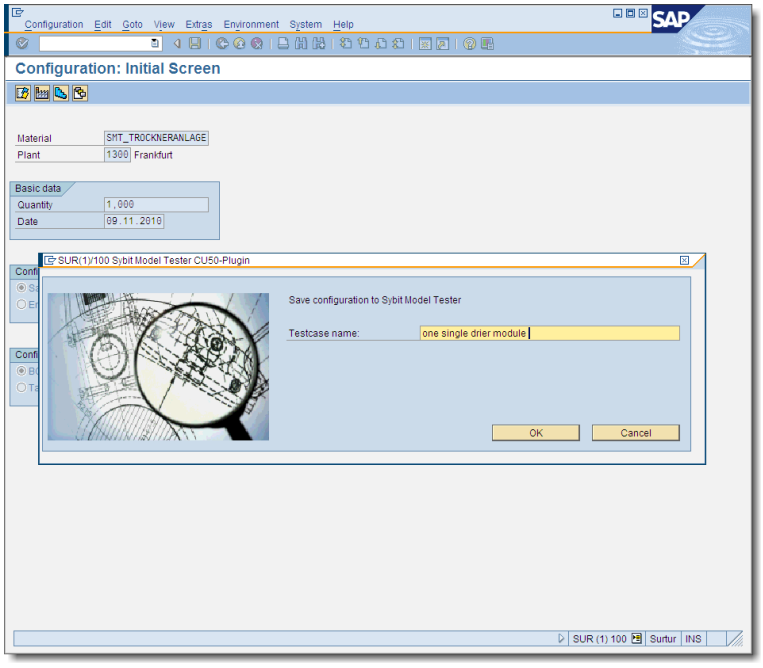


Fig. 4-42: Popup dialogue for saving a configuration as test case in the Sybit Model Tester: Input of test case name.

If you enter a test case name and press OK, the test case is saved in the Sybit Model Tester. If you press Cancel, your configuration will be discarded.

5 Error Handling

This section tells you how you should deal with application errors.

But what is an application error? In principle there are two kinds of errors: Errors that terminate the application (so called dumps) and logical errors that cause the application to behave differently as described in this manual.

5.1 Termination (Dump) of the Application

Fig. 5-1 shows an example of an error that terminates the application. If this occurs, the Sybit Model Tester makes sure that the underlying database is always consistent. To this end, all changes that have been entered after the last click on a save button are dismissed.

Please proceed as follows:

1. Retrieve the system context by clicking on the link **Retrieve context** and save it to your local computer, e.g. your desktop. The saved information is important for error analysis.
2. Open a new email message and address it to support.smt@sybit.de .
3. Append the saved file as an attachment to your email.
4. Note down the steps that caused the error, e.g.: *Loaded test case <NAME> → assigned value <VALUE> to characteristic <CHARACTERISTIC> → saved the test → jumped to Test Run by Quick Link → error occurred.* Please attach this text also to the Email.
5. Restart the Sybit Model Tester by clicking F5.

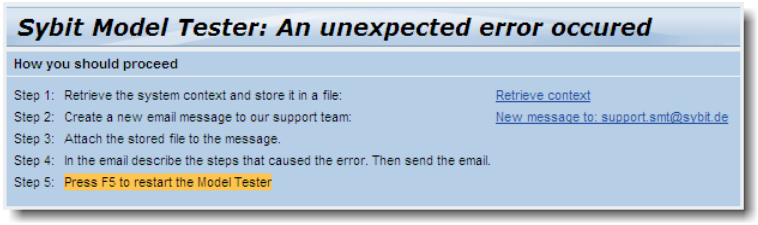

Sybit Model Tester: An unexpected error occured
How you should proceed
Step 1: Retrieve the system context and store it in a file:
Retrieve context
Step 2: Create a new email message to our support team:
New message to: support.smt@sybit.de
Step 3: Attach the stored file to the message.
Step 4: In the email describe the steps that caused the error. Then send the email.
Step 5: Press F5 to restart the Model Tester

Fig. 5-1: Termination of the application.

5.2 Wrong Functionality

An example for wrong functionality could be that before saving a test case a description has been defined. However, on load of the test, this description is not displayed any more.

Proceed as follows:

1. Try to reproduce the behavior, i.e. find out if you can produce the error again and again. To track the error it is helpful, if your reproduction contains as few steps as possible.
2. Note down the steps of your reproduction, e.g.: *Loaded test case <NAME> → assigned value <VALUE> to description → saved the test → edit off → loaded test → description is not set*. Insert this reproduction instruction into an email.
3. Send the Email to support.smt@sybit.de.

Appendix

A. Messages

The tables in this appendix contain all possible messages that may be contained in the messages tab of the expected or factual result of a test case.

A.1. VC-Specific Messages

The following messages are returned by the VC and adopted by the Sybit Model Tester.

Message
Invalid currency
Invalid date
Invalid time
Invalid format
Invalid interval
Interval forbidden
Invalid pattern
Invalid unit of measure
Invalid base measure
Invalid size
Value not found
Value not possible
Change rejected
Value inconsistent
Characteristic univalent
Characteristic not selectable
Characteristic not found
Characteristic locked
Object characteristic
Characteristic not assigned
Value inconsistent
Restricted value domain
Configuration inconsistent
Unknown error
Input too long
Unknown error

A.2. Errors in the Test Case Structure

The following error messages are triggered, if the structure made from header KMat and sub KMats of a multi level test case is inconsistent with the structure proposed by the VC.

Message	Description
Structure conflict	(Deprecated: Not used any more)
Obsolete test structure	The structure of the test case contains a sub KMat, that is not contained in the structure retrieved from the VC.
Test structure not found	The structure of the test cases lacks a sub KMat that has been part of the structure returned by the VC.

A.3. Error Messages Caused by Testing Prices

SMT 2.0

The following error messages are triggered by testing prices in a test run.

Message	Description
Condition not found	For the variant condition and the calculation scheme in the header of the test case, there exists no condition record in all access sequences of all access.
Access sequence inconsistent	For the given variant condition, the system found a condition record in an access of a sequence. However, the variant condition is not found in the last access of the same sequence.

For more information about theses messages, please refer to the section on expected prices in the test case tab, see Section 4.7.8.

B. Further Information

User manual:	www.sybit.com/smt/user_manual.html
Web page Sybit Model Tester:	www.sybit.com/smt
Web page Sybit GmbH:	www.sybit.com
Email address for support inquiries:	support.smt@sybit.de